# SELF-HELP ASSESSMENT AND TRAINING SUPPORT SYSTEM FOR SPECIAL CHILDREN

**MILTON WIDER**

**UNIVERSITI TEKNOLOGI MALAYSIA**

**UNIVERSITI TEKNOLOGI MALAYSIA**

**DECLARATION OF THESIS / UNDERGRADUATE STUDY PAPER AND COPYRIGHT**

Author's full name : MILTON WIDER

Date of birth : 15 JANUARY 1988

Title : DEVELOPMENT OF SELF-HELP ASSESSMENT AND TRAINING SUPPORT SYSTEM FOR SPECIAL CHILDREN

Academic Session : 20010/2011

I declare that this thesis is classified as: √

| | | |
|---|---|---|
| ☐ | **CONFIDENTIAL** | (Contains confidential information under the Official Secret Act 1972)* |
| ☐ | **RESTRICTED** | (Contains restricted information as specified by the organisation where research was done)* |
| ☑ | **OPEN ACCESS** | I agree that my thesis to be published as online open access (full text) |

I acknowledged that Universiti Teknologi Malaysia reserves the right as follows:

1. The thesis is the property of Universiti Teknologi Malaysia.
2. The Library of Universiti Teknologi Malaysia has the right to make copies for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

| | |
|---|---|
| **SIGNATURE** | **SIGNATURE OF SUPERVISOR** |
| 880115-12-5451<br>**(NEW IC NO. /PASSPORT NO.)** | En. Abd Hamid Bin Ahmad<br>**NAME OF SUPERVISOR** |
| Date: 13 May 2011 | Date: 13 May 2011 |

**NOTES :** * If the thesis is CONFIDENTIAL or RESTRICTED, please attach with the letter from the organisation with period and reasons for confidentiality or restriction.

"I/We hereby declare that I/we have read the contents of the thesis and to my/our opinion, the contents of the thesis have fulfilled the scope and quality as required to award the Bachelor's Degree of Engineering (Electrical-Medical Electronics)".

| | |
|---|---|
| Signature | : ..... |
| Name of Supervisor I | : En. Abd Hamid Bin Ahmad |
| Date | : 13 May 2011 |

| | |
|---|---|
| Signature | : ..... |
| Name of Supervisor II | : Assoc. Prof. Ir. Dr. –Ing. Eko Supriyanto |
| Date | : 13 May 2011 |

# SELF-HELP ASSESSMENT AND TRAINING SUPPORT SYSTEM FOR SPECIAL CHILDREN

**MILTON WIDER**

**Submitted to the Faculty of Electrical Engineering**

**in partial fulfillment of the requirements for the award of the**

**Bachelor of Engineering (Electrical-Medical Electronics)**

**Faculty of Electrical Engineering**

**Universiti Teknologi Malaysia**

**MAY 2011**

I declare that this thesis entitle *" Self-Help Assessment and Training Support System for Special Children"* is the result of my own research except as cited in the reference. This thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

| | | |
|---|---|---|
| Signature | : | ……. |
| Name | : | MILTON WIDER |
| Date | : | 13 May 2011 |

Dedicated, in thankful appreciation for support, encouragement and understandings to my beloved family, lecturers and friends.

# ACKNOWLEDGEMENT

First of all, I would like to express my heartily gratitude to God for His grace, wisdom, knowledge, intelligence, strength and patience that He had gave me throughout the process of this projects.

My thanks to my supervisor, En. Abd Hamid Bin Ahmad and my co-supervisor, Assoc. Prof. Ir. Dr. –Ing. Eko Supriyanto for their full guidance, patience and enthusiasm given throughout the progress of this study.

My appreciation also goes to my beloved family who has been so tolerant and supports me all these years. Thanks for their encouragement, love, emotional and financial supports that they had given to me.

I would also like to thank Medical Electronics Laboratory assistant, Mr. Haikal and Mr. Indra as well as others who are not mentioned for their co-operations, guidance and helps in this study.

Nevertheless, my great appreciation dedicated to my friends, GIFTers and SEP & SEB member's batch 2007 and those that involve directly or indirectly with this study. Thanks for continuously giving me moral support, building comments and being a companion throughout these years. Thank you very much.

# ABSTRACT

Children with Down syndrome always associated with mental retardation, specific characteristic of facial appearance and poor muscle build. So, early intervention is a process of providing educations and supports to children to lessen the effect of this disability and delay. However, early intervention program is not widely implemented in Malaysia due to limitation of training centers for special children and lack of human resources with experience training in the field. Hence, Self-help assessment and training support system for the special children is developed. The support system is highly reliable which uses international recognized curriculum as its assessment and training activities. A quantitative measurement is perform for every assessment and generated through a report for record. The process started by taking the child, parents and trainer's detail. Follow by the assessment on self-help skills. A report that shows the child's performance in percentage is generated after the assessment and proceeds with training to improve the child's performance. The result shows that the parents are able to do the assessment and training on their child continuously. As a conclusion, the child's self-help skills are improved by continuous assessment and training.

Keywords: Down Syndrom, international recognized curriculum, self-help skill, assessment, training.

# ABSTRAK

Kanak-kanak istimewa selalu dikait rapat dengan masalah kebantutan minda, mempunyai karakter paras rupa yang tertentu dan otot yang lemah. Oleh itu, program intervensi awal ialah suatu proses untuk memberi pendidikan dan sokongan kepada kanak-kanak yang diberi rawatan kepada fisikal serta tumbesaran terbantut tersebut. Walau bagaimanapun, program intervensi awal tidak banyak diimplimentasikan di Malaysia oleh sebab kekurangan pusat latihan untuk kanak-kanak istimewa serta individu yang berpengetahuan dalam bidang ini. Oleh itu, sistem sokongan penilaian dan latihan terhadap keupayaan berdikari untuk kanak-kanak istimewa telah dicipta. Sistem sokongan ini mengadaptasikan kurikulum antarabangsa yang diiktirafkan sebagai activiti untuk penilaian serta latihan. Pengiraan yang kuantitatif dilakukan pada setiap penilain dan dihasilkan melalui satu laporan sebagai rekod. Proses ini dimulai dengan merekod maklumat kanak-kanak, ibu bapa serta penilai. Kemudian penilain akan dilakukan. Satu rekod yang menunjukkan prestasi kanak-kanak tersebut dalam bentuk peratusan dihasilkan setelah penilaian dilakukan dan diteduskan dengan latihan untuk meningkatkan presstasi kanak-kanak. Keputusan menunjukkan bahawa ibu bapa dapat melakukan penilaian serta latihan secara berterusan. Kesimpulannya, keupayaan berdikari kanak-kanak tersebut meningkat melalui penilain dan latihan yang berterusan.

Kata kunci: Down Syndrom, kurikulum antarabangsa yang diiktiraf, keupayaan berdikari, penilaian, latihan.

# TABLE OF CONTENTS


| CHAPTER | TITLE | PAGE |
|---|---|---|
| | **DECLARATION OF THESIS** | ii |
| | **DEDICATION** | iii |
| | **ACKNOWLEDGEMENT** | vii |
| | **ABSTRACT** | viii |
| | **ABSTRAK** | ix |
| | **TABLE OF CONTENTS** | x |
| | **LIST OF TABLES** | xii |
| | **LIST OF FIGURES** | xiv |
| **1** | **INTRODUCTION** | 1 |
| | 1.1 Background | 1 |
| | 1.1.1 Current Situation | 2 |
| | 1.1.2 Study Comparison | 2 |
| | 1.2 Problem Statement | 3 |
| | 1.3 Objective | 4 |
| | 1.4 Scope | 4 |
| | 1.5 Expected Output | 4 |
| | 1.6 Alternative Solution | 5 |
| | 1.6.1 Available Solution | 5 |
| | 1.6.2 Proposed Solution | 6 |
| | 1.7 Innovation/Novelty | 7 |
| | 1.8 Thesis Outline | 7 |

| | | | |
|---|---|---|---|
| **2** | **LITERATURE REVIEW** | | 9 |
| | 2.1 | Down Syndrome | 9 |
| | 2.2 | Self-help Skills | 10 |
| | 2.3 | Early Intervention Program (EIP) | 13 |
| | 2.4 | Assessment | 14 |
| | 2.5 | Self-help Development | 14 |
| | 2.6 | C# Programming Language | 16 |
| **3** | **METHODOLOGY** | | 18 |
| | 3.1 | Introduction | 18 |
| | 3.2 | Gantt Chart | 19 |
| | 3.3 | System Block Diagram | 22 |
| | 3.4 | Work flow | 24 |
| **4** | **DESIGN AND IMPLEMENTATION** | | 25 |
| | 4.1 | Introduction | 25 |
| | 4.2 | System Configuration | 26 |
| | 4.3 | System Specification | 26 |
| | 4.4 | Function of System | 27 |
| | 4.5 | Subsystem Explanation | 28 |
| | | 4.5.1 Computer Based Processing Unit | 28 |
| | | 4.5.2 Water Level Detection Based input Unit | 29 |
| | 4.6 | List of self-help skills | 30 |
| | | 4.6.1 Self-help skills Subgroup | 31 |
| | 4.7 | System Implementation | 32 |
| | | 4.7.1 Software Implementation | 32 |
| | | 4.7.2 C# Programming Language | 33 |
| | | 4.7.3 Hardware Implementation | 37 |
| | | 4.7.4 Hardware Development and Tools | 37 |

| | | | |
|---|---|---|---|
| **5** | **RESULT AND ANALYSIS** | | 42 |
| | 5.1 | Objective and Scope | 42 |
| | 5.2 | Self-Help Assessment and Training Support System GUI | 43 |
| | 5.3 | Quantitative Measurement | 45 |
| | 5.4 | Functional Testing | 46 |
| | 5.5 | Performance Testing | 49 |
| | 5.6 | Safety Testing | 51 |
| | 5.7 | User acceptance testing | 52 |
| | 5.8 | User Data Management | 56 |
| **6** | **CONCLUSION AND RECOMMMENDATION** | | 58 |
| | 6.1 | Conclusion | 58 |
| | 6.2 | Novelty | 58 |
| | 6.3 | Important result | 59 |
| | 6.4 | Comparison with other methods | 59 |
| | 6.5 | Limitation of solution | 60 |
| | 6.6 | Recommendation | 60 |
| | **REFERENCES** | | 61 |
| | **APPENDICES** | | 63 |

# LIST OF TABLES


| TABLE NO. | TITLE | PAGE |
|---|---|---|
| 2.1 | An overview of developmental progress for typically developing children, from birth to five years. | 10 |
| 2.2 | Achieving independence in self-help skills | 11 |
| 2.3 | Milestones for personal social/ self-help | 12 |
| 2.4 | Feeding, dressing, grooming and toileting skills development of a child | 15 |
| 3.1 | Gantt chart for FYP1 | 19 |
| 3.2 | Gantt chart for FYP2 | 20 |
| 4.1 | List of self-help skills | 30 |
| 4.2 | Self-help skills subgroup for sleeping | 31 |
| 5.1 | Self-help assessment and training support system Layout | 43 |
| 5.2 | Functional testing results | 47 |
| 5.3 | Performance testing results | 48 |
| 5.4 | Safety testing results | 50 |
| 5.5 | User acceptance testing test result | 51 |
| 5.6 | User data management | 53 |

# LIST OF FIGURES


| FIGURE NO. | TITLE | PAGE |
|---|---|---|
| 3.1 | Block diagram of the system | 22 |
| 3.2 | Work flow of the project | 24 |
| 4.1 | Self-help system configuration | 26 |
| 4.2 | Computer based processing unit | 28 |
| 4.3 | Water level detection | 29 |
| 4.4 | Self-help support system main component | 32 |
| 4.5 | Self-help support system flow chart | 33 |
| 4.6 | GUI application source code | 36 |
| 4.7 | GUI application for assessment | 37 |
| 4.8 | Hardware algorithm | 38 |
| 4.9 | Water level detection | 38 |
| 4.10 | Circuit diagram | 39 |
| 4.11 | System configurations for water level detection | 41 |
| 4.12 | Assessment layout for the activity pours the liquid to the cup | 41 |
| 4.13 | Condition of the cup when half full and full | 41 |
| 5.1 | Main page of the self-help assessment and training support system | 43 |
| 5.2 | Patient data input to insert child, parents and trainer details | 44 |
| 5.3 | Assessment session | 44 |
| 5.4 | Report section | 45 |

| | | |
|---|---|---|
| 5.5 | Training session | 45 |
| 5.6 | Result in graphical form | 46 |
| 5.7 | Changes on the layout display | 53 |
| 5.8 | Changes on the font size | 54 |
| 5.9 | Changes on the report presentation | 54 |
| 5.10 | implementation of audio for instruction | 55 |
| 5.11 | Patient data | 56 |
| 5.12 | Self-help database | 57 |
| 5.13 | Self-help assessment and training support system user manual | 57 |

# CHAPTER 1

# INTRODUCTION

## 1.1 Background

Down syndrome is the most common cause of mental retardation and malfunction in a newborn baby. It is a chromosomal disorder caused by an additional chromosome in the 21st pairs of chromosome. Current statistic states that this syndrome occurs one in 800 live births with equal frequency in people of different nationalities, social backgrounds and economic classes [13]. Children with Down syndrome always associated with developmental delay, a specific characteristic of facial appearance, poor muscle tone which influences gross and fine motor development, language development delays which due to muscle problem and cognitive limitation. However, these difficulties and limitations can be overcome and greatly improved by the implementation of Early Intervention Program (EIP).

Early intervention program (EIP) is a systematic program of exercises, therapy and activities constructed to help developmental delays that may be experienced by Down syndrome children or other disabilities [5]. Some services provided in early intervention program are physical therapy, structured simulation program and language therapy. Early intervention program is highly recommended to be introduced to children shortly after birth.

### 1.1.1 Current Situation

To-date, early intervention program is not widely implemented in Malaysia due to limitation of training centers for special children and lack of human resources with experience training in the field [8]. Mostly, the early intervention program is center-based program where the children have to be sent to the program center and being assessed particular times per week. Hence, there is an urgent attention to develop an effective and better early intervention program with a continuous assessment for the special children.

### 1.1.2 Product Comparison

Currently, some government and private organization had been establish a programs to help Down Syndrome Children to improve their disabilities and delays. Most of the program is center based training. It is highly dependent on the trainer's skills and experience on performing an effective assessment and training. Assessment and training is done continuously using suitable exercises to improve the particular activity. The process of assessment and training are generally based on The Hawaiian Learning Profile, BCD Instructional activities and so on. Table 1.1 shows the comparison between the assessment and training resources.

**Table 1.1:** The comparison between different resources for EIP [13]

| **Characteristics** | **HELP** | **BCD** | **Children Laptop** |
|---|---|---|---|
| **Functions** | To assess and provide trainings | To assess and provide trainings | To provide interactive training |
| **Price** | RM 150 per set | RM 180 per set | RM 50 per piece |
| **Advantages** | Highly reliable and proven effective | Highly reliable and proven effective | Able to attract children's attention and lower cost |
| **Limitations** | Highly dependent on trainers' experience and hardly interpreted | Highly dependent on trainers' experience and hardly interpreted | Assessment not included, and no individual training given |
| **Reliability** | High | High | Low |

## 1.2 Problem Statement

There are few modules in ELISSA which aim to lessen the development delays in Down syndrome children namely gross motor, fine motor, social emotional, language, cognitive and self-help. However, some of the modules are incomplete. It is crucial to complete the modules in order to have a sufficient and effective early intervention program to help the Down syndrome children. Self-help assessment and training is one of the incomplete modules. A new self-help assessment and training is needed to have an improved and effective module. Furthermore, early intervention program is hard to be found in Malaysia. This is due to some factors; lack of special children training center, lack of experience trainers and lack of instrument or effect database for assessment. There is also no continuous assessment. Parents are required to bring their child with special needs to the center at least once a week for assessment and training. This is due to the unavailability of software assessment that can be used by parents so that they can do their own assessment and training at home.

## 1.3 Objective

The objectives to be archived are:

- To develop self-help assessment and training database for the special children.
- To implement the self-help assessment and training support system in newly develop software for quantitative measurement.
- To implement a hardware in aiding one of the self-help assessment activity.

## 1.4 Scope

There are several scope outlined in this study to ensure the objectives is achieved. The scopes are:

- This study is limited only to the special children within the age range from 0 to 6 years old.
- Focuses on self-help assessment and training database for special children.
- Software implementation using C# programming language.
- Hardware implementation on eating activity specifically the ability to fill cup with water.

## 1.5 Expected Output

There are several expected outputs at the end of the project. The expected outputs are:

- A new software interface for continuous self-help skills assessment and training for special children.
- A new water level detection hardware to aid in the assessment process.
- An improved software system with more user friendly interaction.

## 1.6 Alternative Solution

### 1.6.1 Available solution

The Orton Gullingham multisensory method [26] which introduced by Anna Gillingham and a group of master teachers. Dr. Samuel Orton assigned Anna's group the task of designing a whole new way of teaching the phonemic structure of our written language to people with dyslexia. In this method they use six steps approach which are simultaneous multisensory instruction, intense instruction with ample practice, direct and explicit Instruction, systematic and cumulative, synthetic and analytic and diagnostic teaching. This method however cannot be performed by commoners instead it needs the specialists who are more familiar with the methods to perform all the activities.

KinderKare an Early Intervention Program serving infants and toddlers ranging from birth to three years of age. they provide Early Intervention therapy and special education services for children and their parents. They provided program to aid on language, cognitive, gross motor, daily life skills and etc. The method that they use is through teaching and learning by experience personals. It is solemnly depend on the experience of the personal who handle the child with the use of some toys for teaching purpose.

All the methods stated as above for early intervention program are depends on the trainers experience and understanding to conduct assessment and training on the

child. There are no specific indications on how much successful rate for the child to master the specified training.

### 1.6.2 Proposed solution

The method that this project proposed is to build a software for assessment and training. It can be use anywhere and anyone. Since the existence early intervention program is highly dependent on the personal experience, the parents have to send their children to the center for assessment and training. This project propose a friendly approach where everyone can use the software to perform the assessment and training even for the nonspecialist.

The existing methods only rely on the trainer's self-evaluation on the child performance which can be misleading because of its unreliability. However, the proposed method uses database that adopted from international recognize curriculum on special children and uses quantitative assessment which can show the learning progress of the child through a chart that generated after the assessment. The software also has the training system with detail instructions to enable the users to perform training on the child themselves. This is more convenient for the parents because they can do the assessment and training on their child themselves. Table 1.2 shows the comparison between the different solutions available and the newly develop support system.

**Table 1.2:** The comparison between current solutions to solve the problems in the current EIP and the newly developed support system

| . **Characteristics** | **HELP/BCD** | **Children Laptop** | **Newly developed support system** |
|---|---|---|---|
| **Functions** | Assessment and training | Training | Assessment and training |
| **Price** | High | Medium | Low |
| **Advantages** | Highly reliable and proven effective | Able to attract children's attention and lower cost | Highly reliable and proven effective |
| **Limitations** | Highly dependent on trainer's experience | Assessment not included | Partially dependent on trainer's experience |
| **Reliability** | High | Low | High |

## 1.7 Innovation/Novelty

There are some novelties in this project. The novelties are:

- A whole new early intervention program software support system interface
- A Special features implemented such as voice instruction when operating the system for easier operational experience
- New method on generating report which generate pie chart that shows the assessment in percentage.
- New application on self-help assessment and training for special children.

## 1.8 Thesis Outline

This thesis consists of six chapters. It introduces the details on the development of the self-help assessment and training support system for children with special needs.

The first chapter is the introduction. In this chapter discussed about the background study, problem statement the objectives of the study as well as the scope of the study.

Chapter 2 shows the literature review of the study. First of all it explains the details on Down syndrome. Physical features and poor cognitive ability of the child is discussed especially on the self-help skills. The early intervention program that will help to improve the child learning progress also been discussed in this chapter. Finally, the introduction of assessment and C# programming language that is used for this project are discussed.

Chapter 3 is the methodology of the study. Here shows the gantt chart, system block diagram and work flow of the study.

Chapter 4 is about design and implementation. In this chapter, the self-help system configuration is explained in details. Then a list of self-help database is shown. The database is divided into 4 sub-groups namely eating, dressing, toileting and sleeping. Software implementation also discussed here; the software algorithm and flow chart for the self-help support system. Finally, the water level detection algorithm for the hardware implementation is also discussed in this chapter.

Chapter 5 is about result and analysis. In this chapter, the GUI display for the self-help assessment is displayed. The finish product of the water level detection also is displayed. A user acceptability test is discussed. User's feedback is discussed in detail for better functional support system.

Finally, chapter 6 is about the conclusion and recommendation. A detail explanation is discussed here about what have been done and achieved for the project. Recommendations for the project also are stated for future development.

# CHAPTER 2

# LITERATURE REVIEW

## 2.1 Down syndrome

Down syndrome is a chromosomal disorder caused by an error in cell division that results in extra 21$^{st}$ chromosome [1]. It is associated with mental retardation, a characteristic facial appearance, and poor muscles build [4]. A lot of research shows that 90% of chromosomal disorders in cell division or abnormal cells are the eggs [2]. Maternal age is the only factor that known to affect the probability of having Down syndrome children [8]. For mothers less than 30 years old, one in 1,000 pregnancies will results in a baby with Down syndrome. One in 35 pregnancies in mothers with 44 years old will result in a baby with Down syndrome.

The physical features and medical problems associated with Down syndrome may be varied from child to child. While some needs a lot of medical attention, others live a healthy lifestyle. Kids with Down syndrome tend to share some physical features such as flat facial profile, an upward slant to the eyes, small ears, and protruding tongue [11]. One common characteristic that restrict the development of Down syndrome children is low muscle tone or hypotonia [3]. Most of these children will have significantly delayed spoken language [7] and self-care skills like feeding, dressing and toileting [11].

## 2.2 Self-help skills

Self-help is the ability to take care of everyday personal and social needs [9]. In cases for Down syndrome children, self-help skills are needed to improve their quality of life. Acquiring self-help skills is one of the best ways to help children into independence. In the beginning of birth to 5 years, children remarkably develop skills; communicate in spoken language, walk, run, climb stairs, feed, dress, and go to toilet independently [7]. The table below shows an overview of developmental progress for typically developing children from birth to five years.

**Table 2.1**: An overview of developmental progress for typically developing children, from birth to five years.

| | Birth | 1 year | 2 years | 3 years | 4 years | 5 years |
|---|---|---|---|---|---|---|
| | Family → | | Social world | Preschool | | School |
| **Social** | eye contact/smiling | one-to-one games | sclitary play | play with peers | | being in a group |
| **Motor** | reaching/grasping holding/manipulating | rolling/sitting/standing/walking | | climbing stairs colouring | running drawing | jumping writing |
| **Speech & language** | smiling, cooing, babbling, words | | | keyworc phrases | sentences and grammar | |
| **Self-help** | feeding | using cup | spoon fork | knife | dressing | toileting |
| **Behaviour** | mportance of routines | parents in control | | learning self-control | able to conform co-operate and comply with requests | |
| **Number** | experience of counting | | | | | learning to count |
| **Reading** | experience of books and stories | | | | | learning to read |
| **Memory & thinking** | attending | copying | choosing | remembering events and activities | | learning and thinking skills |

The table below shows the achieving independence in self-help skills for typically developing children. The self-help skills are divided into meals, toileting, washing/ personal care, motor skills, dressing, practical/social. There are several milestones that have been listed out for each of the self-help skills. These milestones have specific range of time that a child is usually able to master.

**Table 2.2**: Achieving independence in self-help skills [9].

| **Meals** | |
|---|---|
| Feeds self with biscuit | 10 months (range 6 -14m) |
| Drinks from a cup | 20 months (range 12 - 30m) |
| Uses spoon and fork | 20 months (range 12 -36m) |
| Can make a sandwich | 50% at 11 - 20 years |
| Can make a cup of tea/coffee | 46% at 11 - 20 years |
| Can heat a can of beans | 28% at 11 - 20 years |
| Can use microwave | 24% at 11 - 20 years |
| Can lay and clear table | 80% at 11 - 20 years |
| Can eat in a restaurant | 100% at 11 - 20 years |
| **Toileting** | |
| Dry during the day | 36 months (range 18 - 50m) |
| Bowel control | 36 months (range 20 - 60m) |
| Dry at night | 60% at 7 - 14 years |
| Use toilet/potty without help | 4 to 5 years |
| Fully continent, day and night | 98% at 11 - 20 years |
| **Washing/personal care** | |
| Washes unaided | 60% |
| Can brush hair | 95% |
| Does brush own hair | 45% |
| Can wash hair | 34% |
| Can run/fill bath | 50% |
| Can cut toe and fingernails | 9% |
| **Motor skills** | |
| Walks alone | 23 months (range 13 - 48m) |
| Climbs/descends stairs unaided | 81 months (range 60 - 96m) |
| Can ride a bike | 35% at 11 - 20 years |
| Can throw and catch a ball | 98% at 11 - 20 years |
| **Dressing** | |
| Dresses self partially (not fastenings) | 4 - 5 years |
| Dresses without help | 80% at 11 - 20 years |
| Chooses appropriate clothes | 70% at 11 - 20 years |
| **Practical/Social** | |
| Can tell the time | 20% at 11 - 20 years |

| | |
|---|---|
| Can count simple amounts of money | 40% at 11 - 20 years |
| Can go to shop alone | 26% at 11 - 20 years |
| Crosses road alone | 40% at 11 - 20 years |
| Note: Mean age of achievement in preschool years or percentage who has achieved the skill in teenage years in Hampshire studies. | |

Children with Down syndrome make progress in all area of development in the same way as the other children but at a slower pace [7]. Although milestones in independence will be achieved in a later age for children with Down syndrome, evidence shows that most young people with Down syndrome do achieve a high level of autonomy in personal care [9]. It is important to stress that for every children with Down syndrome, they will be individual differences in rate of progress towards self-sufficiency. This means that the developmental progress is vary as widely for every Down syndrome child as the typically developing children. The table below shows the milestones for personal/ self-help comparison between children with Down syndrome and typically developing children.

**Table 2.3**: Milestones for personal social/ self-help [7].

| **PERSONAL SOCIAL/ SELF HELP** | | | | |
|---|---|---|---|---|
| **ACTIVITY** | **Children with Down syndrome** | | **Typically developing children** | |
| | **Average age** | **Range** | **Average age** | **Range** |
| Smiles when touched and talked to | 2m | 1.5 to 4m | 1m | 1m to 2m |
| Smiles spontaneously | 3m | 2m to 6m | 2m | 1.5m to 5m |
| Recognises mother/father | 3.5m | 3m to 6m | 2m | 1m to 5m |
| Approaches image in mirror | 6.5m | 4m to 10m | | |
| Takes solids well | 8m | 5m to 18m | 7m | 4m to 12m |
| Feeds self with biscuit | 10m | 6m to 14m | 5m | 4m to 10m |
| Plays pat-a-cake, peep-bo games | 11m | 9m to 16m | 8m | 5m to 13m |
| Holds up arms and legs when getting dressed and undressed | 15m | 12m to 20m | | |
| Drinks from a cup | 20m | 12m to 30m | 12m | 9m to 17m |

| | | | | |
|---|---|---|---|---|
| Uses spoon or fork | 20m | 12m to 36m | 13m | 8m to 20m |
| Urine control during the day | 36m | 18m to 50m+ | 24m | 14m to 36m |
| Plays social/interacting games | 3.5 to 4.5 years | | | |
| Bowel control | 36m | 20m to 60m+ | 24m | 16m to 48m |
| Dresses self partially (not buttons/laces) | 4 to 5 years | | | |
| Uses toilet or potty without help (using a special step) | 4 to 5 years | | | |

## 2.3 Early Intervention Program (EIP)

Early intervention program is a systematic program of therapy, exercise or activities designed to address the developmental delay of children with special needs [5]. The purpose of early intervention is to lessen the effect the disability and delay. Other than that, it is to provide support for families in supporting their children's development, to promote children's development in key domains such as mobility and communication, and to promote children's coping conficence, and to prevent the emergence of future problems [13]. Services, exercise and activities are designed to identify and meet a child's needs in five developmental areas, including: physical development, cognitive development, communication, social or communication development and adaptive development [6].

Early intervention should begin as early as possible. The first years of life are critical time for a child's development. During these early years, all young children go through rapid development that achieve basic physical, cognitive, language, social and self-help skills that set as a foundation for future development progress [5]. Families that participated in the New York state early intervention program one of the largest early intervention systems in United State [17] reported that the early intervention services had helped the families to know their rights, effectively communicate their child's needs, and help their child develop and learn.

## 2.4 Assessment

Assessment includes several steps of collecting data on a child's development and learning, determining its significance goals and objectives of the program, incorporating the information into planning for individuals and programs, and informing the findings to the parents and other involved parties [18]. Assessment is done according to the stages of ages in a child. Infant tests and pre-schools measure different components of intellectual ability. Infant tests measures the developmental progress of babies and children focusing on areas such as gross motor, fine motor, language development, adaptive behaviors and personal-social behaviors [19].

Self-help assessment is the attempt to study the ability to take care of personal daily needs. Personal daily needs can be included such as toileting, dressing and feeding. A measurement tool must be sensitive, save and should be able to offer the assessor full control over the data collection process [25].

## 2.5 Self-help development

Development describes the growth throughout the lifespan of a human from birth to death. The scientific study of human development seeks the understanding on the changes of man lifestyles. This includes all aspect of human growth including physical, emotional, intellectual, social, perceptual, and personality development [20]. Self-help development in a child is a process of learning and mastering skills from birth to the end of adolescence such as sitting, walking, talking and etcetera [21].

Children develop skills as they grow up. They tend to do thing on their own and drive to be independence which is a healthy part of a child development and should be encouraged. There are some main areas that child care provider can foster the

development of self-help skills such as feeding, dressing, grooming, toileting and helping with toilet chore [22].

**Table 2.4**: Feeding, dressing, grooming and toileting skills development of a child [22].

| **Feeding skills** | **Dressing, grooming and toileting skills** |
|---|---|
| **8 months**<br>Gums solid foods<br>Feeds self crackers<br>Holds own bottle<br>Drinks from cup held by adult<br><br>**12 months**<br>Picks up small pieces of cereal to eat<br>Holds cup and drinks with some spills<br><br>**24 - 30 months**<br>Drinks from a small cup, holding it with one hand<br>Bites well through food<br>Brings spoon to mouth with the palm facing up<br>Can eat a meal using a spoon with minimal spilling<br><br>**30 - 36 months**<br>Pours from a small cup<br>Begins to use a fork to spear food<br>Can spread with a dull butter knife<br><br>**3 - 4 years**<br>Pokes food with a fork<br>Begins using fork more than spoon<br>Holds spoon in fingers with palm up<br>Learns to cut soft foods with the edge of fork<br>Learns how to prepare a bowl of dry cereal with milk<br>Uses napkin<br>Serves self at table with little spilling | **15 - 24 months**<br>Learns to take off hat, socks, and shoes without laces<br>Learns to unzip large zippers<br><br>**24 - 30 months**<br>Can pull pants down with assistance<br>Is beginning to unbutton large buttons<br>Anticipates the need to use the toilet<br>Can put shoes on with help<br>Still needs help getting pullover shirts overhead and pants up over bottom<br><br>**30 - 36 months**<br>Puts on items such as shoes, socks, pants, shirts, and jackets<br>Pulls pants up<br>Undresses and dresses with adult help<br>Still needs help with tying<br><br>**3 - 4 years**<br>Uses toilet with adult help<br>Has daytime toilet control<br>Learns to undo fasteners, such as snaps and laces<br>Distinguishes between the front and back of clothes<br>Zips front-opening clothes<br>Still needs adult help with dressing<br>Buttons large buttons<br>Puts on socks, may need help with heel placement<br>Zips and unzips jackets but needs help with starting<br>Puts shoes on, may have trouble telling which feet<br><br>**4 years**<br>Puts on socks with heel placement |

| | |
|---|---|
| | Puts on shoes with some adult help<br>Buckles shoes and belts<br><br>**5 years**<br>Put shirts on correctly<br>Puts belt in loops<br>Unties a tie on an apron<br>Some can put shoes on correct feet and tie laces |

## 2.6 C# programming language

C#, pronounce as C sharp, is a programming language designed for building a wide range of application that run on the .NET framework (*MSDN website*) [8]. It is highly expressive but simple and easy to learn. C# is attended to be simple, modern, general purpose, object oriented program language [23]. The syntax of C# is easy recognisable by those who familiar with C, C++ and JAVA. C# syntax simplifies many of the complexities of C++ and provides powerful features such as nullable value types, enumerations, delegates, lamda expressions and direct memory access, which are not found in JAVA [13].

Originally language C is widely used. C++ came about to add object orientation to language C and become the language of building "real" applications for Windows and used for writing the infrastructure as well as low level application. C# brings rapid development with some obvious changes by taking advantage of the .NET framework [24].

C# gives the capability to build durable system-level components by virtue of the following feature:

- Full COM/platform support for existing code integration.
- Robustness though garbage collection and type safety.

- Security provided through intrinsic code trust mechanisms.
- Full support of extensible metadata concepts.

C# is the simplification of C++ program language syntax to be more consistence and logical while removing some of the more complex feature in C++ language [24]:

- As a type-safe language, C# does not allow direct memory manipulation, so pointers are no longer needed in C#.
- Headers file also have been removed in C# language.
- The namespace and reference operators,:: and -> respectively have been replaced with single operator, the period (.).
- The biggest change is that the *int* and *bool* data types are now completely different.
- Removes memory management issues from the developer by using .NET's garbage collection scheme.

# CHAPTER 3

# METHODOLOGY

## 3.1 Introduction

This chapter will discuss about the progress of this study is undergone throughout the period of time. The system block diagram, work flow of the study as well as the gantt chart of the study will be displayer and explained in detail. Basically, software development and hardware implementation will be done initially. After that, experiment and testing will be conducted to the special children. Lastly, troubleshooting and optimization will be performed for the support system improvement.

Table 3.1 shows the gantt chart for FYP1 and table 3.2 show the gantt chart for FYP2.

## 3.3 System block diagram

The figure below shows the block diagram of the system.

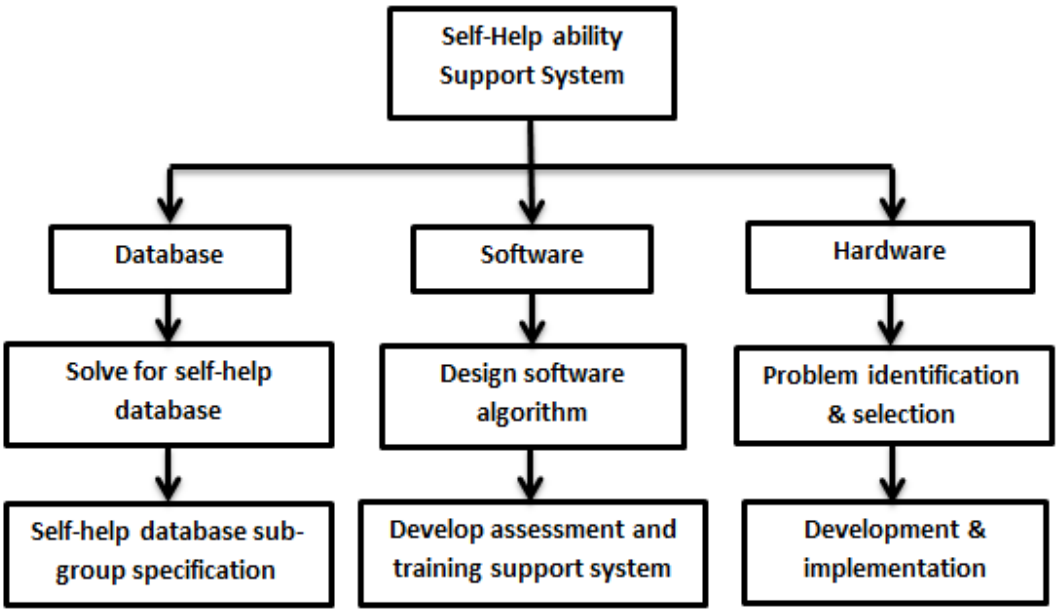



**Figure 3.1** Block diagram of the system

This project divided into of three main parts which are database, software and hardware.

The project is mainly focus on the database construction as well as the software development. The lists of database are constructed based on the international recognized curriculum. It is carefully selected and sorted so that it is relevant to the scope of the study. C# programming language is used to develop the software. A GUI is constructed for the self-help support system. The database is implemented into the support system for effective assessment and training to the special children. Hardware is developed to aid the trainers in doing evaluation for the assessment.

Database is the main component in this study. Firstly, a list of self-help database is identified. A lot of effort had been made to construct a list of self-help assessment. International recognized curriculum is taken as reference for the database to make it reliable and acceptable by the trainer. The database is then specified into four sub-groups namely eating, dressing, toileting and sleeping. The database is essential for better assessment and training for the special children. Thus enable trainers and parents to effectively identify the weakness of the children. Hence, concentrate on giving training to them to improve on the identified action.

Next is software development. First of all, the software algorithm is designed. After that, self-help assessment and training support system is developed. The self-help assessment and training support system is constructed and displayed as GUI. A graphic user interface (GUI) is developed for human-computer interface which can be control by keyboard and mouse. The software that used to develop the support system is C# programming language. The main components for the algorithm are main page, child data input, assessment, report and training.

Last but not least, the implementation of the hardware in the self-help support system. It starts with problem identification and selection. An activity in the self-help skills is chosen based on its practicability to be implemented in hardware to help in doing evaluation by the trainer. Lastly, it ends with the hardware development and implementation. This hardware is to aid the trainers in evaluation when undergoing the self-help assessment on the child.

### 3.4 Work flow

Figure 3.2 shows the work flow of this project.

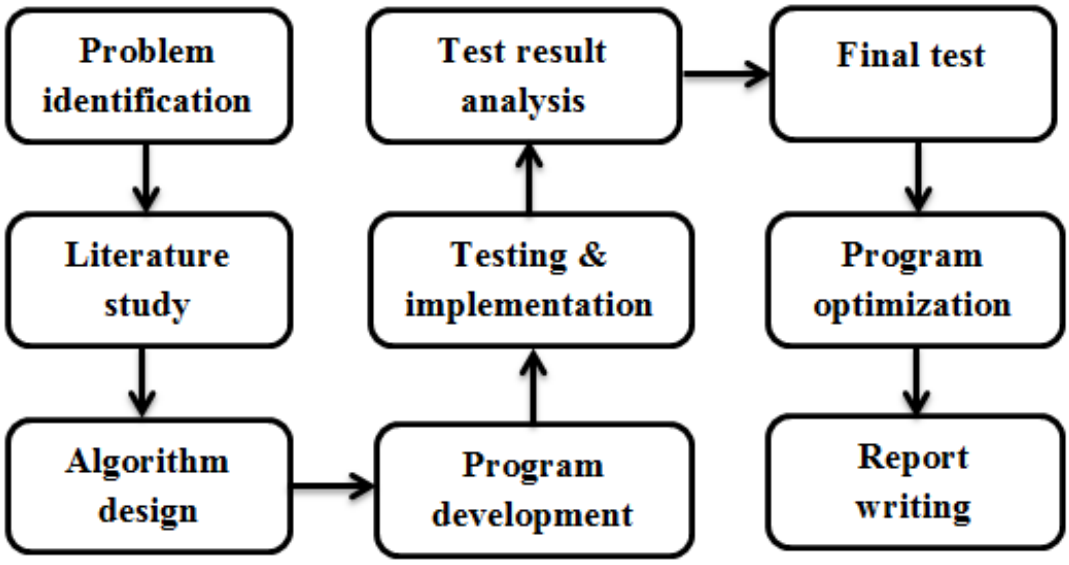



**Figure 3.2** Work flow of the project

The overall progress in this project is summarized in figure 3.2. Firstly, problem is identified followed by literature study on related field through relevant sources such as journals, books, articles as well as conference papers. Then, an algorithm is designed and the software program is developed. After that, test is undergone and continued with result analysis. A final test will be performed for its functionality. Finally, system optimization will be conducted to improve the system and ended with a report.

## 3.2 Gantt chart

Table below shows the gantt chart for this project.

**Table 3.1**: Gantt chart for FYP1

| Task | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Problem Formulation | | | | | | | | | | | | | | |
| Methodology | | | | | | | | | | | | | | |
| Literature Review | | | | | | | | | | | | | | |
| Algorithm | | | | | | | | | | | | | | |
| Circuit design | | | | | | | | | | | | | | |
| Learning software | | | | | | | | | | | | | | |
| Experiment | | | | | | | | | | | | | | |
| Measurement /Test | | | | | | | | | | | | | | |
| Preliminary Results | | | | | | | | | | | | | | |
| Presentation | | | | | | | | | | | | | | |
| Report | | | | | | | | | | | | | | |

**Table 3.2**: Gantt chart for FYP2

| Task | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Hardware implementation | | | | | | | | | | | | | | | | | | | | |
| Software programming | | | | | | | | | | | | | | | | | | | | |
| Testing | | | | | | | | | | | | | | | | | | | | |
| Optimization | | | | | | | | | | | | | | | | | | | | |
| Presentation | | | | | | | | | | | | | | | | | | | | |
| Thesis | | | | | | | | | | | | | | | | | | | | |

# CHAPTER 4

# DESIGN AND IMPLEMENTATION

## 4.1 Introduction

Self-help is the ability to take care of everyday personal and social needs. Before implementing the support system, a list of self-help skills is constructed. This list is important for the assessment and training for the children with Down syndrome. Through this list, trainers or parents can evaluate the progress of the children by assessing their ability to perform as required in the list.

From the list constructed, the skills are grouped in several sub groups, namely eating, dressing, toileting and sleeping. This grouping is essential for a better assessment method. Moreover, it is to ensure that parents and trainers will not have confusion when assessing the child development.

Table shows the list of the self-help skills and table shows the sub group of the self help skills.

## 4.2 System Configuration

Figure 4.1 shows the configuration of the self-help assessment and training support system. The design provides a simple and effective method for self-help assessment and training. The interaction between the trainer and the child will act as the medium to the assessment and training evaluation at the computer. A quantitative evaluation will be display as percentage in a chart from in GUI based on the trainer's evaluation on the child's performance.

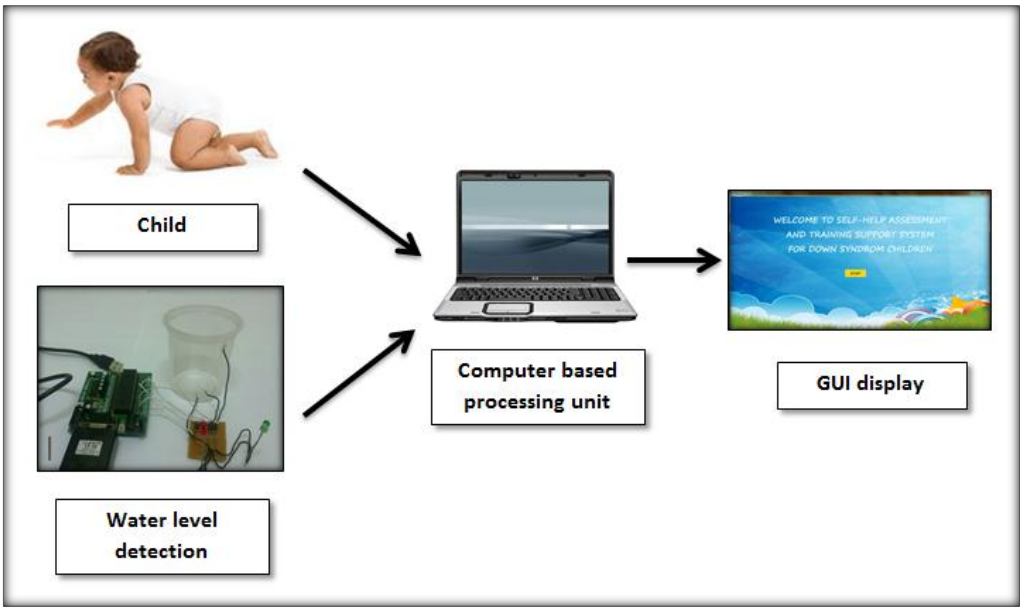



**Figure 4.1** Self-help system configuration

## 4.3 System Specification

The self-help assessment and training support system have some specifications. The specifications are:

**Table 4.1:** System specification

| **Specification** | **Description** |
|---|---|
| Computer operating system (OS) | The self-help assessment and training support system software is perform on Microsoft Window 7 when doing the project. However, the finish product is also tested to other OS namely Microsoft Window XP and Microsoft window Vista. The result shows that it is compatible and can be used on all the OS tested. |
| User | The aim of this project is to be used by the parents with Down Syndrome child so that they can perform continuous assessment and training. However, the user is not limited to the specified individual. Experience trainers, doctors as well as student can use the support system. |
| Child | Self-help assessment and training support system are design to perform on children within the age of 0 to 6 years old. |
| voltage | The water level detection hardware is design to perform at its optimum performance with 9V. However, functioning test have been conducted. It can be operated on 3V, 6V and 12V. |

## 4.4 Function of System

The self-help assessment and training support system can be divided into 2 subsystems. First is the computer based processing unit system and the second is water level detection based input unit system.

First of all, the computer based processing unit uses the database that integrated inside the support system as its main part. Every assessment and training that will be done is highly related to the database. Through this database, the trainer performs

evaluation on the child and the result is generate as a report in the form of pie chart for the trainer record.

The water level detection based input unit system use the water level detection hardware as its main part. These main components of the hardware consist of green and red LED, a cup, PIC and RS232 USB converter. Water level detection is a tool to help in the self-help assessment mainly for eating ability, specifically the ability of the child to pour liquid to the cup. The principle of the water level detection is simply the electric conductivity of the water. The child will pour water in side of the cup, when red LED lit on it indicates that the cup is empty whereas when green LED lit on it indicates that the cup is full.

## 4.5 Subsystem Explanation

The self-help assessment and training support system can be divided into 2 subsystems. The subsystems are computer based processing unit and water level detection input unit. Detail explanation as follow.

### 4.5.1 Computer Based Processing Unit

C# programming language will be implemented as the processing unit inside the computer. Child's performance and trainer's evaluation as the input to the computer processing unit and self-help assessment, training and report will be displayed as an output in term of GUI form. At the input, trainer will register the child, parents as well as the trainer's detail for record. Other than that, the assessment and training also act as the input for the system. Trainer can perform assessment and training on the child through the specified skills base on the detail instruction that available inside the system.

In computer processing unit, a quantitative measurement is perform. This process is coded and implemented using C# programming language. The calculation is coded so that the result will show in percentage. In fact, all the interface between windows inside the GUI is coded using C# programming language. As for the output, a GUI is display on the monitor screen. This GUI provides a user friendly interface for the user. More specifically, the assessment result and training instruction is display.

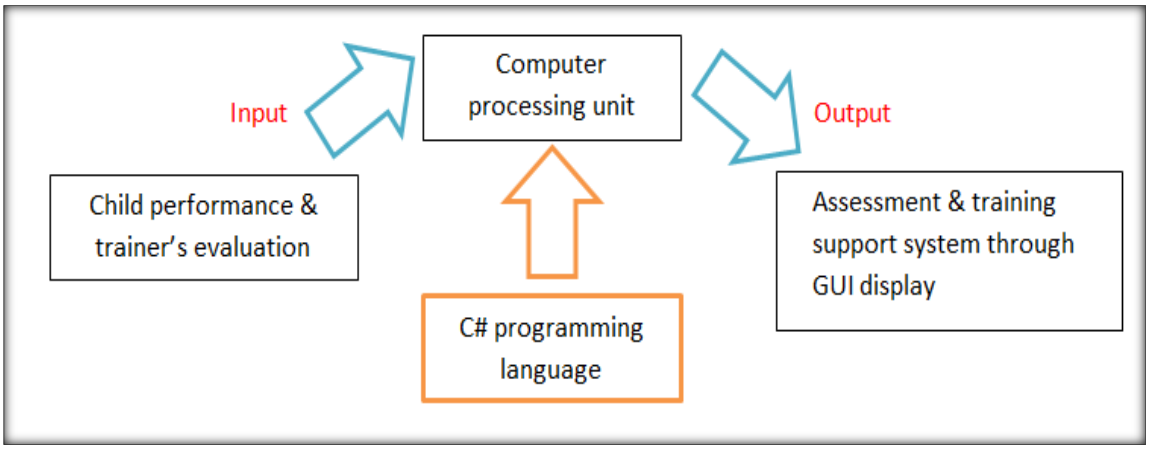



**Figure 4.2** Computer based processing unit

### 4.5.2 Water Level Detection Based input Unit

A water level detection will be implemented in the self-help assessment and training support system to aid one of the assessment activity specifically the activity pour liquid to the cup. Figure 4.3 shows the water level detection that will be used to assist the assessment. The main components that used in the water level detection are a green and red LED, a PIC, RS232 converter and a cup. At the input, water level detection is used as the interaction medium between the system and the user. The LED will light on if the water level is sufficient to allow the electric conduct between the wires. The voltage conducted through the LED is recorded and its behaviour is coded using the C# programming language. Lastly, the behaviour of the LED is shown on the GUI for user's evaluation.

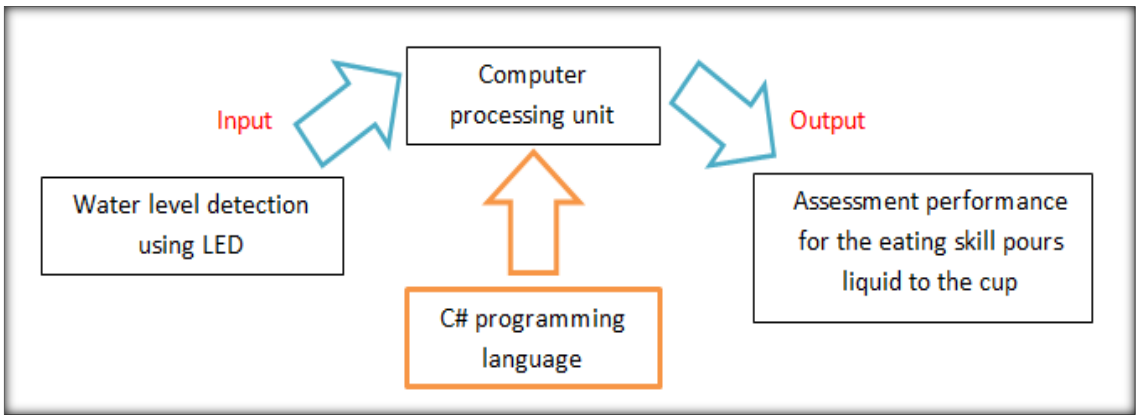



**Figure 4.3** Water level detection based input unit

## 4.6 List of self-help skills

Below show some of the self-help skills. Full list of self-help skill can be found at Appendix C as attach at the end of this thesis.

**Table 4.1:** List of self-help skills

| no | Action | Description |
|---|---|---|
| 1 | Responses to food stimulus | Child opens and closes mouth in response to food stimulus |
| 2 | Sucks and swallows reflex are inhibited | Childs is not able to do the actions of suck and swallow |
| 3 | Brings hand to mouth | Child is able to bring hand to the mouth independently |
| 4 | Recognizes bottle visually | Child is able to recognize bottle visually from other things |
| 5 | Coordinates sucking, swallowing, and breathing | Child is able to suck on bottle or nipple appropriately |
| 6 | Brings hand with toy or object to mouth | Child brings hand with toy or object to the mouth |
| 7 | Pats bottle | Child pats the bottle using hands |
| 8 | Places both hands on bottle | Child places both hands on the bottle appropriately |
| 9 | Swallows strained or pureed foods | Child swallows strained food which are fed to him/her |
| 10 | Rooting reflex is inhibited | Child is not able to perform rooting |
| 11 | Mouths and munches solid foods | Child mouths and gums solid food |
| 12 | Bites and chews toys | Child brings the toys to the mouth and bites them |
| 13 | Uses tongue during eating | Child uses tongue during eating and before the foods are swallowed |
| 14 | Feeds self a cracker | Child holds and feed self a cracker |
| 15 | Holds own bottle | Child holds own bottles independently |
| 16 | Bites food voluntarily | Child bites the foods voluntarily during eating |
| 17 | Drinks from cup held for him | Child is able to drink from cup held for him correctly |
| 18 | Drools less except when teething | Child drools less except when teething |
| 19 | Feeds self with fingers | Child feeds himself/herself using hands and fingers |
| 20 | Holds spoon | Child is able to hold the spoon independently |

### 4.6.1 Self-help skills Subgroup

Self-help skills are divided into 4 subgroups namely eating, dressing, toileting and sleeping. Table below only show the self-help subgroup skills for sleeping. Full list of subgroups skills can be found at Appendix C as attach at the end of this thesis.

**Table 4.2:** Self-help subgroup skills for sleeping

| **Sleeping** | | |
|---|---|---|
| **No** | **Action** | **Description** |
| 1 | Sleeps ten-fifteen hours daily | Child sleeps ten-fifteen hours daily |
| 2 | May awake and cry from dreams | Child may awake and cry from his/her dreams |
| 3 | May refuse naps | Child may refuse naps sometimes |
| 4 | Sleeps four-ten hour intervally | Child sleeps four-ten hour with intervals at night |
| 5 | Stays awake for long periods without crying at night | Child stays awake for long periods without crying at night |
| 6 | Naps frequently | Child naps frequently in a day |
| 7 | Sleeps ten-twelve hours with night awakening | Child sleeps ten-twelve hours with night awakening |
| 8 | Naps one-four hours, two-three times each day | Child naps one-four hours. two-three times each day |
| 9 | Sleeps twelve to fourteen hours | Child sleeps twelve to fourteen hours each day |
| 10 | Naps one-four hours, once or twice each day; may refuse morning nap | Child naps one-four hours, once or twice each day; child may refuse napping in the morning |
| 11 | Sleeps ten-twelve hours | Child sleeps ten-twelve hours a day |
| 12 | Naps one-three hours once in afternoon | Child naps around one-three hours once in the afternoon |
| 13 | Delays sleeping by demanding things | Child tends to delay sleeping by demanding things |

## 4.7 System Implementation

The system implementation can be divided into two parts which are software implementation and hardware implementation.

### 4.7.1 Software Implementation

The software algorithm can be explained with the flow chart below. Figure 4.4 indicates the main component that needed to be achieved in the support system. Self-help support system should be able to insert child data input before performing assessment on the child. After assessment are finish trainer can see the report in report section and lastly perform training for the child to improve the self-help skills.

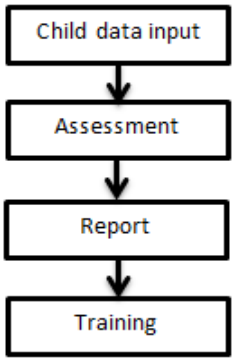



**Figure 4.4** Self-help support system main component

The figure 4.5 shows the detail flow chart for a functional self-help support system. Trainers start at the main page where they are needed to insert child the information. If the child is already registered in the database, the trainer can select the child data and display it to see its information and proceed to the assessment session. If the child is a new patient then the trainer need to insert the child, the child parents and

the trainer information in the form specified. Next, the trainer can start the assessment on the child based on the self-help skills database provided. After that, trainer can view the child's progress in the report section based on the assessment evaluation. The trainer can continue to perform training at the training session to improve the child self-help skills.

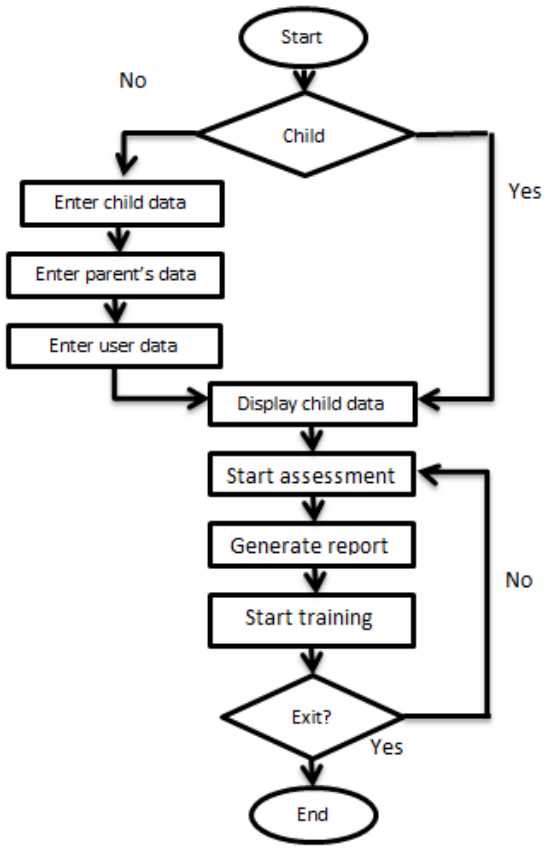



**Figure 4.5** Self-help support system flow chart

### 4.7.2 C# Programming Language

C# programming language is used for the development of self-help assessment and training support system. Microsoft Visual Studio software is used to utilise the C# programming language that available inside the software. C# programming language is

used due to its vast and variety of libraries that able to achieve the objective for the development of the self-help support system.

The source code below shows the C# programming language that used to develop the software. The display source code below is the code written for the software main page. The full source code can be found at the Appendix A attach at the end of this thesis.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace SelfHelpGUI
{
    public partial class MainForm : Form
    {
        public MainForm()
        {
            InitializeComponent();
        }

        private void start_Click(object sender, EventArgs e)
        {
            welcomeheader.Visible = false;
            mainheader.Visible = true;
            start.Visible = false;
            patientdatainput.Visible = true;
            assessment.Visible = true;
            report.Visible = true;
            training.Visible = true;

            System.Media.SoundPlayer sp = new
System.Media.SoundPlayer("C:\\Users\\Milton\\Documents\\Visual Studio
2010\\Projects\\SelfHelpGUI\\SelfHelpGUI\\Sound\\welcome2.wav");
            sp.Play();
        }

        private void patientdatainput_Click(object sender, EventArgs e)
        {
            patientdatainputform patientdatainputform = new
patientdatainputform();
            patientdatainputform.Show();
```

```
        }

        private void assessment_Click(object sender, EventArgs e)
        {
            toiletinglabel assessmentform = new toiletinglabel();
            assessmentform.Show();
        }

        private void report_Click(object sender, EventArgs e)
        {
            reportform reportform = new reportform();
            reportform.Show();
        }

        private void training_Click(object sender, EventArgs e)
        {
            trainingform trainingform = new trainingform();
             trainingform.Show();
        }

        private void MainForm_Load(object sender, EventArgs e)
        {
            System.Media.SoundPlayer sp = new
System.Media.SoundPlayer("C:\\Users\\Milton\\Documents\\Visual Studio
2010\\Projects\\SelfHelpGUI\\SelfHelpGUI\\Sound\\welcome.wav");
            sp.Play();
        }
    }
}
```

Some explanation on the code:

```
System.Media.SoundPlayer sp = new
System.Media.SoundPlayer("C:\\Users\\Milton\\Documents\\Visual Studio
2010\\Projects\\SelfHelpGUI\\SelfHelpGUI\\Sound\\welcome2.wav");
            sp.Play();
```

This code is use to play the voice instruction when starting the software. The code search the specified file at the directory and play it.

```
patientdatainputform patientdatainputform = new patientdatainputform();
patientdatainputform.Show();
```

This code identify a new specified window and open it if click.

The source code is written to generate the GUI for the self-help assessment and training support system. The main GUI is the main page, patient data, assessment, training and report display. In patient data, there are 3 other GUI which is child data, parent's data and trainer's data.

Figure 4.6 below shows the actual workspace in Microsoft Visual Studio 2010 where the code is written. Figure 4.7 shows the finish GUI after the code is correctly written.

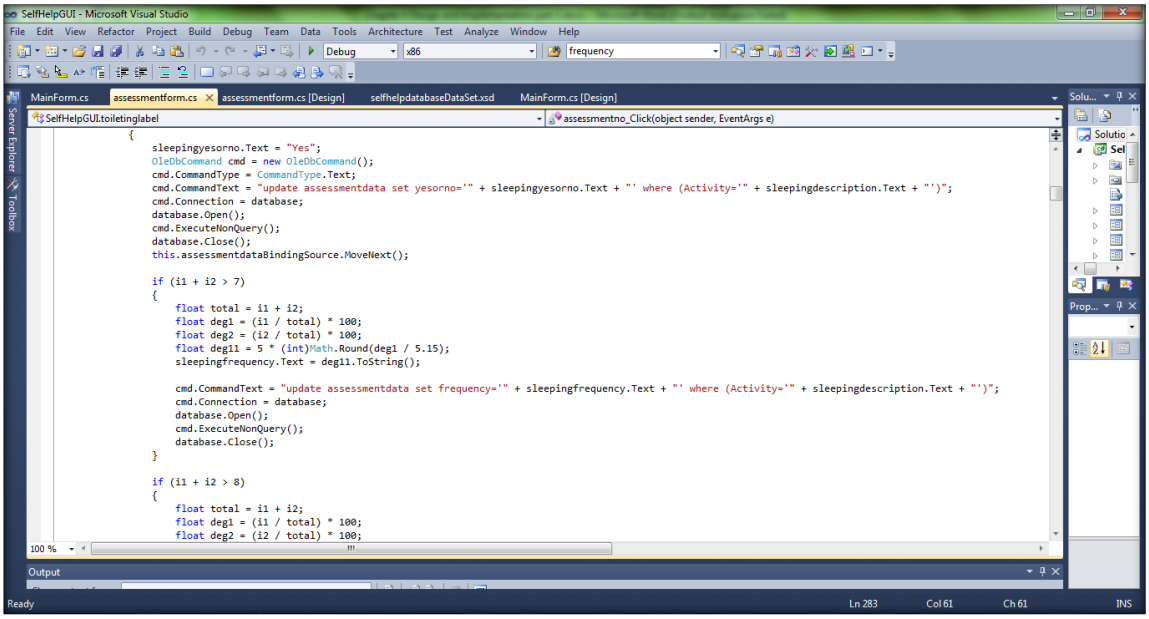



**Figure 4.6** GUI application source code

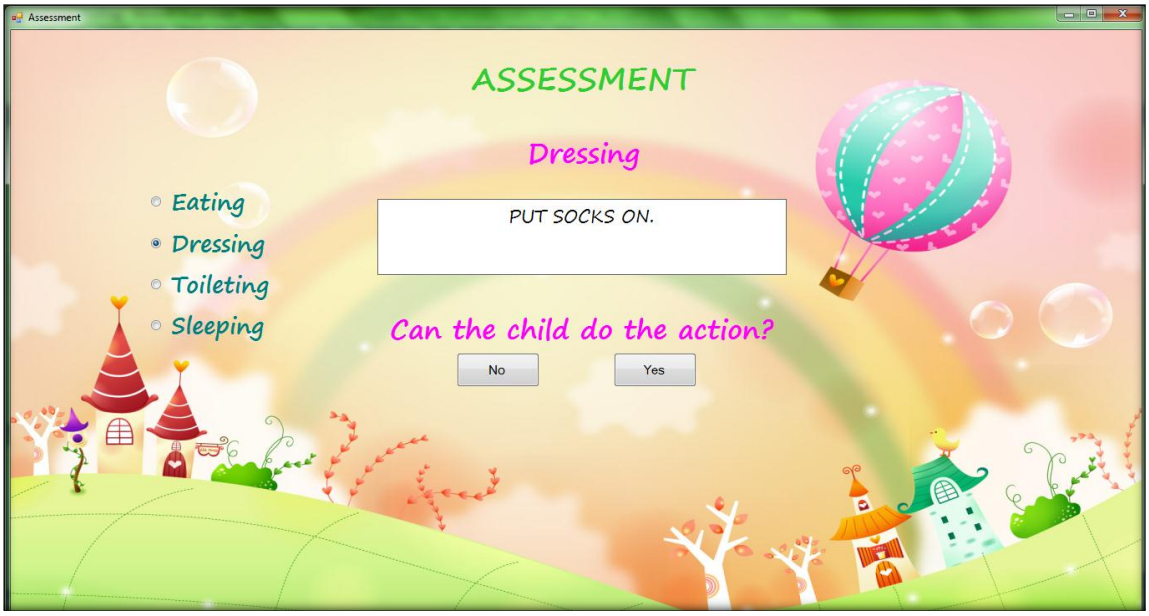



**Figure 4.7** GUI application for assessment

### 4.7.3 Hardware Implementation

The implementation of the support system is crucial to ensure the effectiveness and efficiency of assessment and training delivery to the child. Water level detection is implemented to the support system to aid in the child training. A list of self-help assessment skills is used in the assessment and training session to guide the trainer to assess and train the important skills that the child needs to learn.

### 4.7.4 Hardware Development and Tools

Water level detection is used to help in eating skills for the activity pour liquid to the cup. Figure 4.8 shows the algorithm for the main component of the water level detection. LED is used to detect the conductivity of the water in the cup between the

wires. PIC is program to detect the changes of the LED state. RS232 converter will send the data to processor and display it on the monitor.

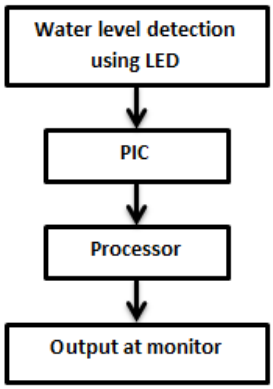



**Figure 4.8** Hardware algorithm

Hardware for water level detection had been developed to assist in the evaluation for the eating skill assessment specifically the activity to pour liquid into the cup. Figure 4.9 shows the finish product of the water level detection hardware. This hardware will detect the water level that pours into the cup that by the child. The water level detection uses the principle of electricity conductance in water. A common wire will be place at the bottom of the cup along with the red LED wire. Another wire for the green LED is place at the top of the cup.

Hence when water is fill in the cup and the water level touches the common with the red LED wire, the red LED will lit on. This indicates that the cup is half full and the child requires to fill the cup with more water. When the water level touches the green LED wire, the green LED will lit on. This in turn indicates that the cup is full with water. The child is successfully undergo the assessment for pouring liquid into the cup and can proceed to do for the next assessment.

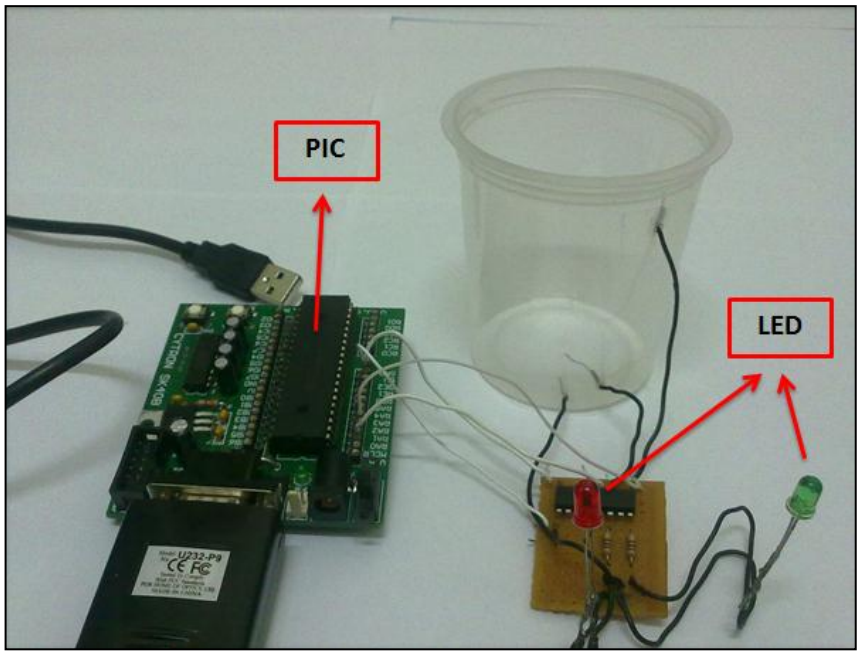



**Figure 4.9** Water level detection

Water level detection uses electric conductivity in the water as its principle for water detection. Below shows the circuit diagram of the hardware.

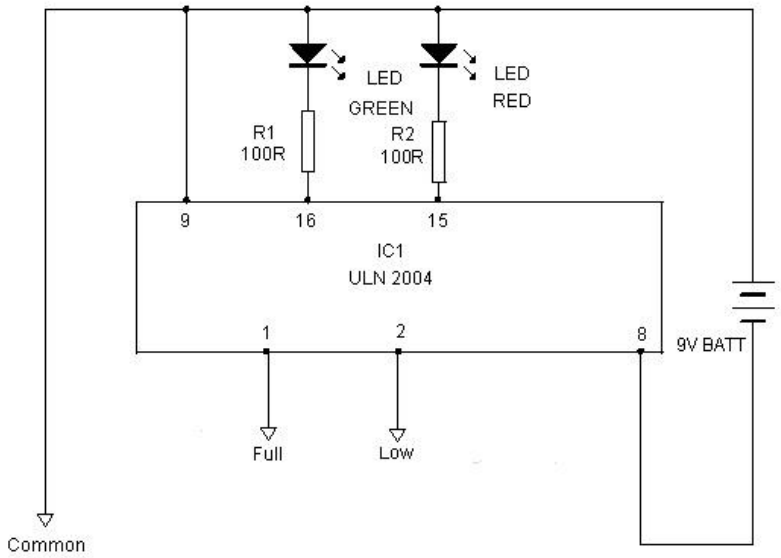



**Figure 4.10** Circuit diagram

Figure 4.10 shows the circuit diagram of water level detection. Some components are used for the design namely IC ULN2004, green and red LED, two resistors with value 100 ohm and a 9V battery. A 9V battery or adaptor can be used as the hardware power source. The common, full and low points are put inside the cup. The common and low point is put at the beneath of the cup whereas the full point is put near top of the cup. This design uses the electric conductance principle. When the points inside the cup are connected when the water is pour inside the cup, the electric will travel through the water and make a close circuit which enable the LED to light on. Hence, when water is pour into the cup until it reach the common and low point, the red LED will light on indicating that the water level is low or the cup is not full. When water is pour until the water level reaches the full point, the green LED will light on indicating that the cup is full with water.

The water level detection is connected to the computer by RS232 converter. A PIC was programmed to detect the LED states from no light to light on state and send the information to the computer through RS 232 converter. Figure 4.11 shows the system configuration for the water level detection.

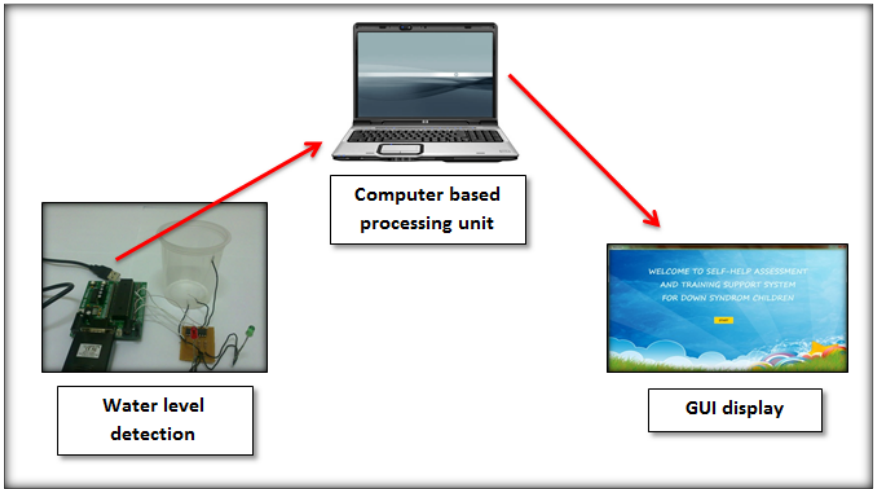



**Figure 4.11** System configurations for water level detection

The lit on LED in water level detection send the information to the computer based processing unit that coded with C# programming language and display the result in self-help assessment and training support system GUI in assessment session for eating skills specifically the activity pours liquid to the cup. Figure 4.12 shows the layout of the assessment for the activity pour the liquid into the cup. Figure 4.13 shows the condition when the cup is half full and full. When the red LED lit on the black panel in the picture will lit red whereas when the green LED lit on the upper black panel will lit green as shown in the picture.

**Figure 4.12** Assessment layout for the activity pours the liquid to the cup

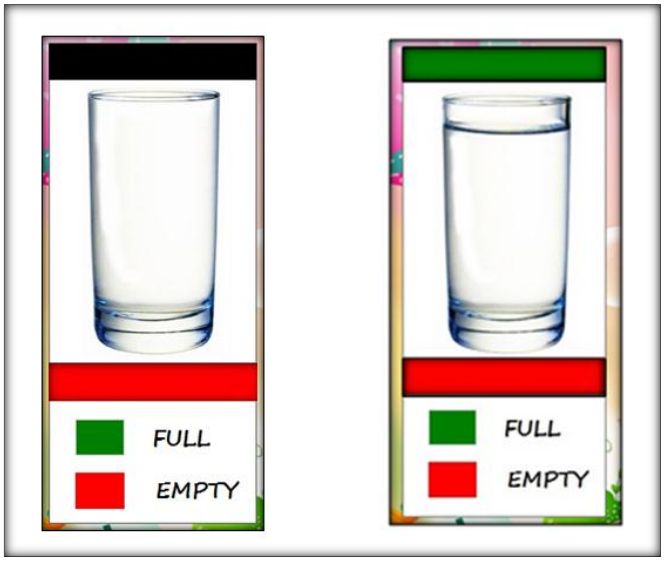



**Figure 4.13** Condition of the cup when half full and full

# CHAPTER 5

# RESULT AND ANALYSIS

## 5.1 Objective and Scope

In this chapter, the result of self-help assessment and training support system GUI, water level detection hardware, quantitative measurement, user acceptability test as well as user data management will be displayed.

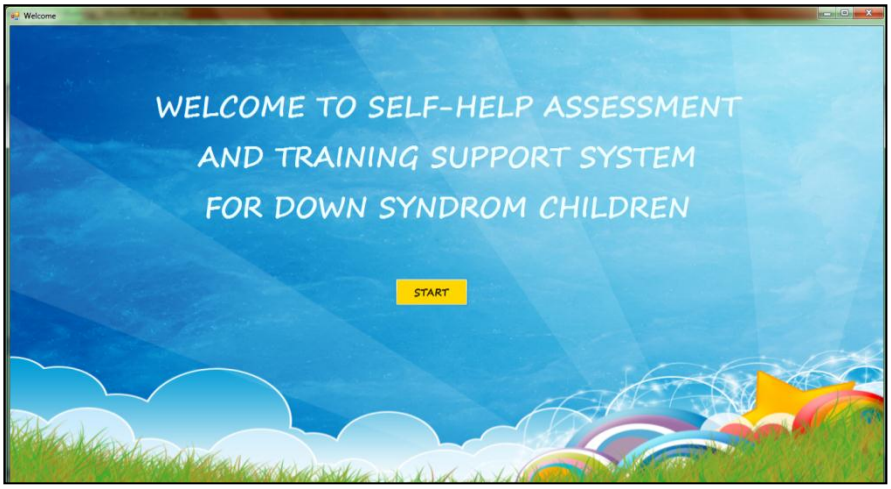
The objective of the analysis is sub system testing, functional testing and to perform a user acceptance testing for user's feedback on the self-help assessment and training support system. This is necessary to identify the reliability and functionality of the support system. The scope for testing the support system can be classified by who will perform the test and by whom it will be tested.

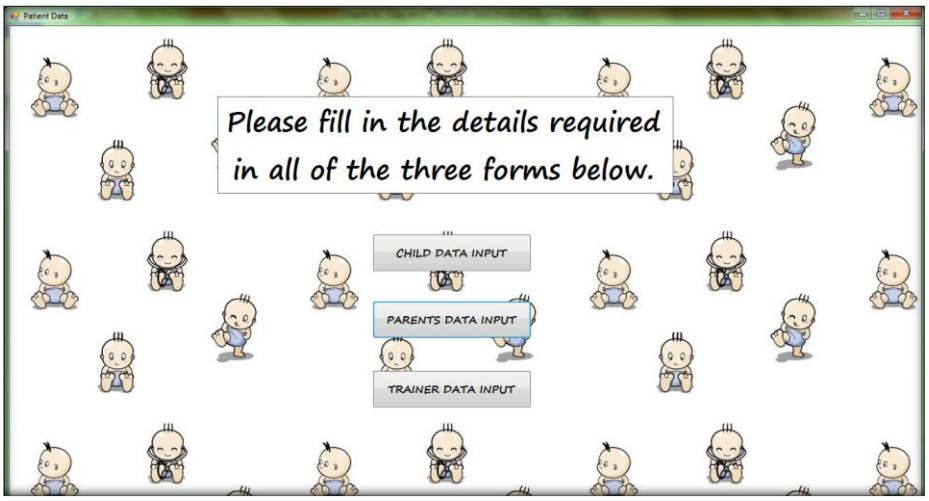
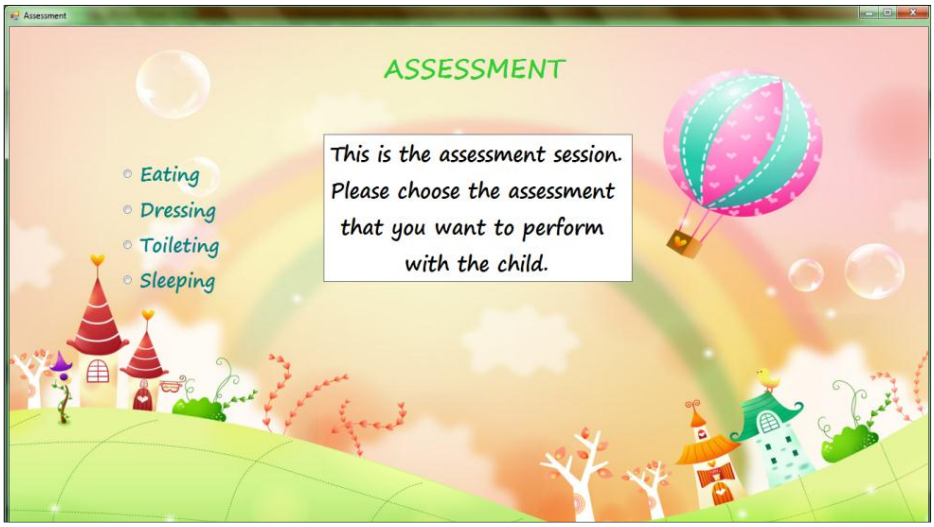
The people who can perform the testing cover a wide range of individual. It can be performed by trainers, students, parents and etc. This flexibility also implemented to the test subject. This means that the test subject can be a normal child, child with special needs, adolescent or grown up individual.

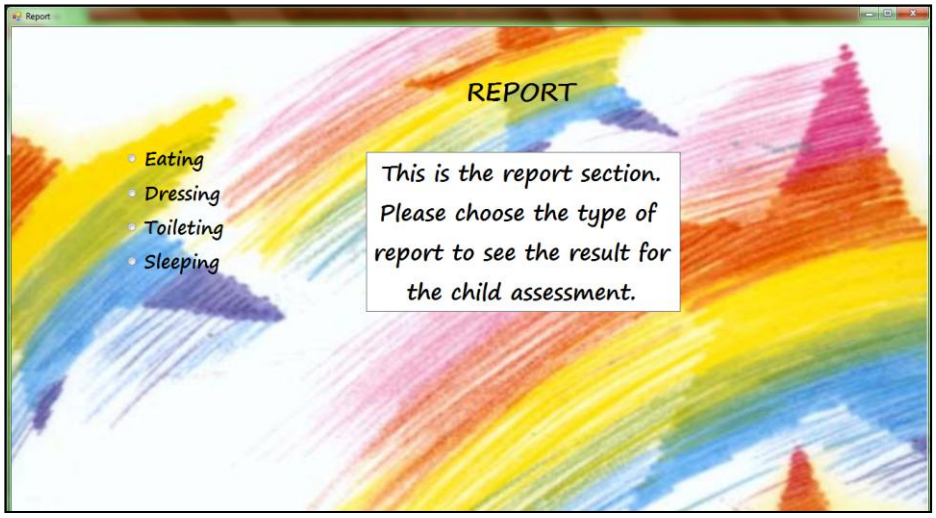
## 5.2 Self-Help Assessment and Training Support System GUI

Table 5.1 shows the layout of the support system. Trainer begins the self-help assessment and training from the main page. First of all the user need to fill in the child, parents and the trainer details in the specified form inside the support system. Next, trainer can navigate to the assessment section to do self-help assessment specifically eating, dressing, toileting and sleeping. After finish each of the assessment, trainer can see the report of the assessment at the report section. This report will show the progress of the child self-help assessment in percentage. Finally, trainer can perform training on the child at the training section to increase the ability of the child self-help skills.

**Table 5.1**: Self-help assessment and training support system layout

| **Items** | **Results** |
|---|---|
| Main page |  **Figure 5.1** Main page of the self-help assessment and training support system |

| | |
|---|---|
| Patient data input |  <br>**Figure 5.2** Patient data input to insert child, parents and trainer details |
| Assessment |  <br>**Figure 5.3** Assessment session |

| | |
|---|---|
| Report |  <br>**Figure 5.4** Report section |
| Training | <br>**Figure 5.5** Training session |

## 5.3 Quantitative Measurement

The result is display in graphical form. The graphical form is construct base on the trainer's evaluation in the assessment session. Each self-help skills assessment will produce different chart to show the progress of the child.

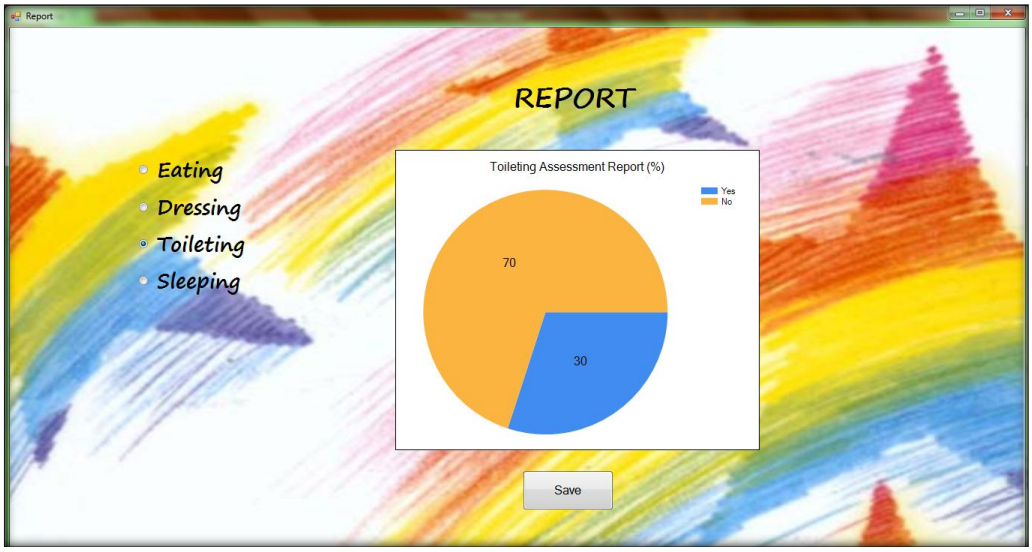



**Figure 5.6** Result in graphical form

For a perfect score in self-help assessment, the percentage is 100% for every assessment. Hence trainer has the obligation to perform training and assessment on the child to reach as high percentage as possible. Higher percentage means that the child is capable of doing the specified self-help skills whereas lower percentage indicates a limitation on performing the skills. Thus a training session can be performed to increase the capability of the child.

## 5.4 Functional Testing

Functional testing was performed to test for the self-help assessment and training support system functionality. The functional testing was done in two parts which are for the computer based processing unit or the software and the water level detection or hardware. The functional testing results were shown below.

**Table 5.2:** Functional testing results

<table>
<tr><th>Test</th><th>Results</th></tr>
<tr><td>Computer operating system</td><td>
Table 5.3: Operating system and its compatibility
<table>
<tr><th>Operating system</th><th>Compatibility</th></tr>
<tr><td>Windows 7</td><td>Yes</td></tr>
<tr><td>Windows XP</td><td>Yes</td></tr>
<tr><td>Windows Vista</td><td>Yes</td></tr>
</table>
The software and hardware was tested to different type of computer operating system. The objective is to test the compatibility of the software with different type of OS. The Operating systems that were used for testing were Windows7, Windows Vista and Windows XP. The result shows that the software was compatible with all of the operating systems.
</td></tr>
<tr><td>Database updates</td><td>The software was tested when updates were performed on the database. Database can be updated by editing, adding or deleting the database inside the Microsoft Access where the data was stored. The result shows that the software was functioning accordance to the changes that made on the database.</td></tr>
<tr><td>Type of liquid</td><td>
Table 5.4: Liquid and its conductivity
<table>
<tr><th>Liquid</th><th>Conductivity</th></tr>
<tr><td>Tap water</td><td>Yes</td></tr>
<tr><td>Drinking water</td><td>Yes</td></tr>
<tr><td>Orange juice</td><td>Yes</td></tr>
</table>
The hardware was tested to different type of liquid. The objective is to test for the electric conductivity inside the cup with different type of
</td></tr>
</table>

<table>
<tr><td></td><td>liquids. The liquids that were used for the testing were tap water, drinking water and orange juice. The result shows that the hardware was functioning well with all of the liquids.</td></tr>
<tr><td>Power source</td><td>The hardware was tested to different kind of power source namely battery and power adapter. The objective is to test for its functionality for different kind of power source. The result shows that the hardware was functioning for both power sources.</td></tr>
<tr><td>Voltage supply</td><td>Table 5.5: Voltage and its functionality<br><table><tr><th>Voltage (V)</th><th>Functionality</th></tr><tr><td>3</td><td>Yes</td></tr><tr><td>6</td><td>Yes</td></tr><tr><td>9</td><td>Yes</td></tr><tr><td>12</td><td>Yes</td></tr></table><br>The hardware was tested to different voltage as power supply. The objective is to test for its functionality for different voltage. Adapter with adjustable voltage was used with adjustable value voltage of 3V, 6V, 9V and 12V. The result shows that the hardware was functioning for all of the different voltage.</td></tr>
</table>

Through the functional testing, it can be concluded that the self-help assessment and training support system can function well in all of the specified test. Since any changes on the database did not affect the functionality of the support system, the user can update the database if needed according to the child's requirement for assessment and training. The users do not have to be worried if there was no electricity to supply to the hardware because it can be operated by using portable battery. The hardware was designed to have a 9V as the input for its optimum performance. However, different voltage also can be used for it to function. A thorough functional testing can be done if needed for future improvement on its functionality.

## 5.5 Performance Testing

Performance testing was performed to test for the self-help assessment and training support system performance. The performance testing was done in two parts which are for the computer based processing unit or the software and the water level detection or hardware. The performance testing results were shown below.

**Table 5.6:** Performance testing results

<table>
<tr><th>Test</th><th>Results</th></tr>
<tr><td>Operating duration for software</td><td>
Table 5.7: Operating duration for software and its functionality
<table>
<tr><th>Hours</th><th>Functionality</th></tr>
<tr><td>3</td><td>Yes</td></tr>
<tr><td>6</td><td>Yes</td></tr>
<tr><td>9</td><td>Yes</td></tr>
<tr><td>12</td><td>Yes</td></tr>
</table>
The software was tested by opening the support system for a given time duration. The support system was left open for a total of 12 hours duration without supervision. The support system was tested every 3 hours. The result shows that the software was functioning as usual even open after a long time.
</td></tr>
<tr><td>Operating duration for hardware</td><td>
Table 5.8: Operating duration for software and its functionality
<table>
<tr><th>Hours</th><th>Functionality</th></tr>
<tr><td>3</td><td>Yes</td></tr>
<tr><td>6</td><td>Yes</td></tr>
<tr><td>9</td><td>Yes</td></tr>
<tr><td>12</td><td>Yes</td></tr>
</table>
</td></tr>
</table>

| | |
|---|---|
| | The hardware was tested by turning on the power supply to the hardware for a given time duration. The water level detection was left open for a total of 12 hours duration without supervision. The hardware was tested after every 3 hours duration. The result shows that the hardware was functioning as usual even open after a long time. However, the performance of the hardware was low if battery was used as the power source in this performance test. |
| Operation on computer under heavy load | The software was tested for computer with heavy load with small storing of memory left. A computer with heavy load such as using antivirus and other software that used a lot of space inside the computer was used. The software was install inside the computer and tested as usual. The result shows that the software can function to perform assessment and training. However, loading the software when starting the support system was slow. Generating the report after each assessment also was slow. |
| Power source | The hardware was tested to different kind of power source namely battery and power adapter. The objective is to test for its performance for different kind of power source. The result shows that the hardware was functioning for both power sources. However, the light intensity of the LED was low when using battery compared to the light intensity when using the power adapter. |
| LED | Both of the LED was tested for its performance. The LED was left light on for a duration of time. A total of 1 hour was used for the performance test. The result shows that the LED was still light on even after the duration time ended. However, the surface of the LED was hot after the test. |

Through the performance testing, it can be concluded that the self-help assessment and training support system have a high performance under different kind of

situations. The support system as well as the water level detection can be operated for a long time but still will be at its best performance. The water level detection has a quite low maintenance since the LED also can be operated for a long time. However, it is advisable that the LED is not used for too long because it might burn the device from overheating. A thorough performance testing can be done if needed for future improvement.

## 5.6 Safety Testing

Safety testing was performed to test for the self-help assessment and training support system safety over the user and the child. The safety testing was done only to the water level detection or hardware. The safety testing results were shown below.

**Table 5.9:** Safety testing results

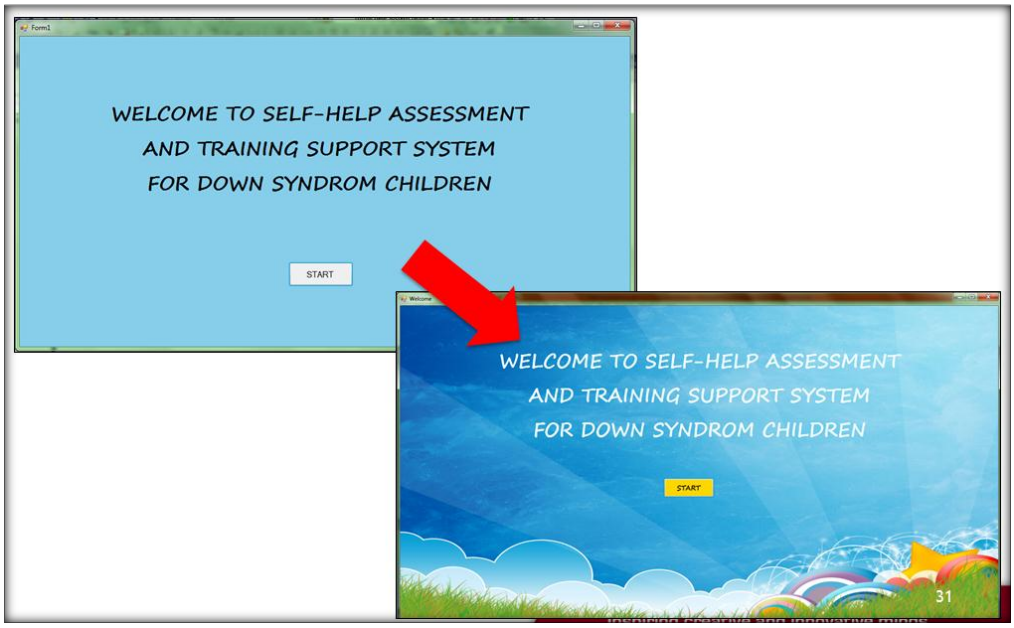
| **Test** | **Results** |
|---|---|
| Voltage | The hardware was tested to different voltage as power supply. The objective is to test for its safety for overload. Adapter with adjustable voltage was used with adjustable value voltage of 3V, 6V, 9V and 12V. The result shows that the hardware was functioning for all of the different voltage. |
| Liquid inside the cup | The hardware was tested to different type of liquid. The objective is to test for the safety of the trainer and child over the liquid inside the cup that used for assessment. The liquids that were used for the testing were tap water, drinking water and orange juice. The water was felt with fingers during assessment was performed. The result shows that there was no uncomfortable sensation when the liquids were exposed to the skin. |

Through the safety testing, it can be concluded that the self-help assessment and training support system is safe to be used by the trainer and over the child. A different voltage can be used to operate the hardware. The trainer do not need to be afraid over electric shock when performing assessment using water level detection because it do not do any harm to the body. However, it is advisable that the user should always turn off the power supply if the water level detection is not used. The liquid that used for the assessment also is advisable not to drink or give the child to drink. It is strictly only for assessment and training purpose. The water should be thrown away after the assessment or training. A scientific testing can be performed for future improvement on the system's safety.

## 5.7 User acceptance testing

The method of testing is through hands on testing. This means that an amount of tester is selected, for this case a total of 10 testers, to use the support system for assessment and training. Every comment are recorded and analysed for its relevancy to the support system. Table 5.5 shows the user acceptability test result, feedback from the user for functional test. The feedbacks included layout display, font size, report presentation and audio instruction.

**Table 5.10**: User acceptance testing test result

| Events | Results |
|---|---|
| Layout display |  <br>**Figure 5.7** Changes on the layout display<br><br>User's feedback: the layout is dull and uninteresting to attract the children attention for the self-help assessment and training.<br>Solution: The dull layout is change to a more colourful and interesting layout. |

Font size

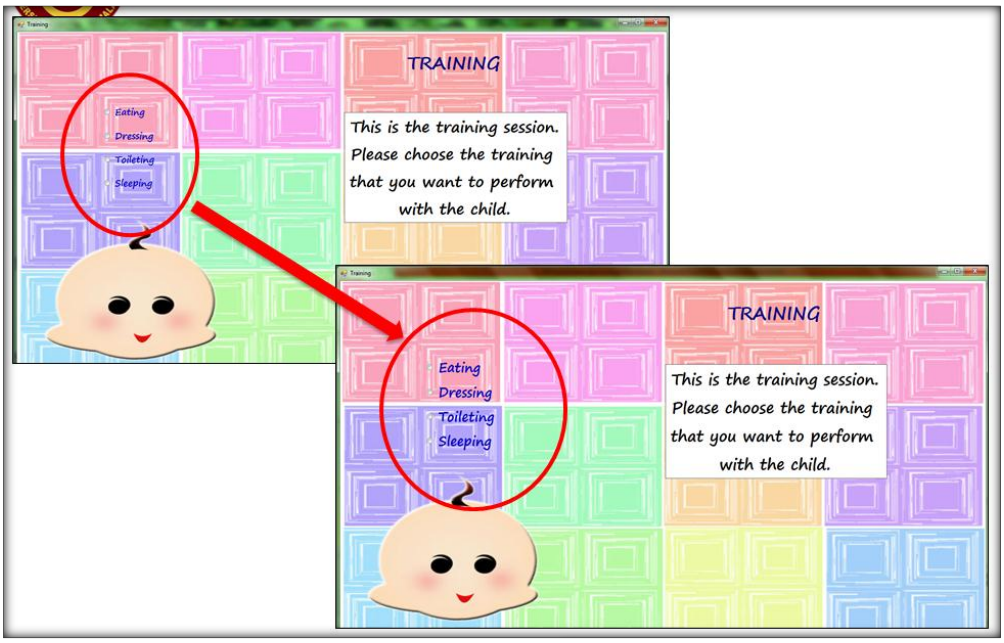


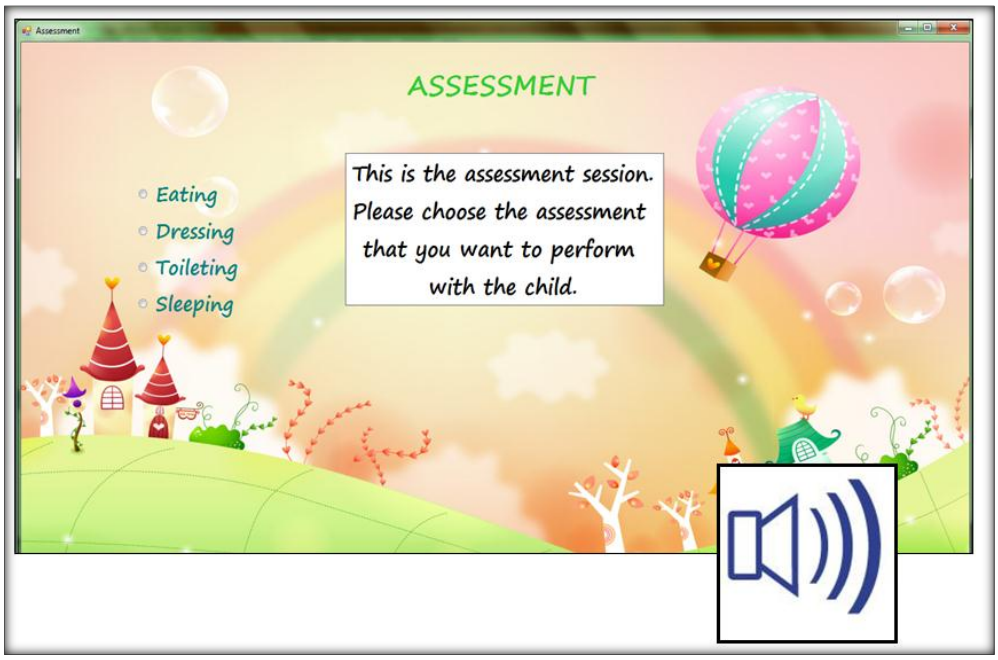

**Figure 5.8** Changes on the font size

Use's feedback: The font is too small and hard to be read by the user.

Solution: The font is change to bigger and more visible size to be read.

Report layout

**Figure 5.9** Changes on the report presentation

| | |
|---|---|
| | User's feedback: The user wanted to know the progress of the child by percentage.<br>Solution: the layout of the report is change to percentage. |
| Audio |  <br>**Figure 5.10** implementation of audio for instruction<br><br>User's feedback: User faced difficulty on operating or using the self-help assessment and training support system without the user manual.<br>Solution: Audio is implemented in the support system to give a clear instruction for assessment and training from the beginning and throughout the session. |

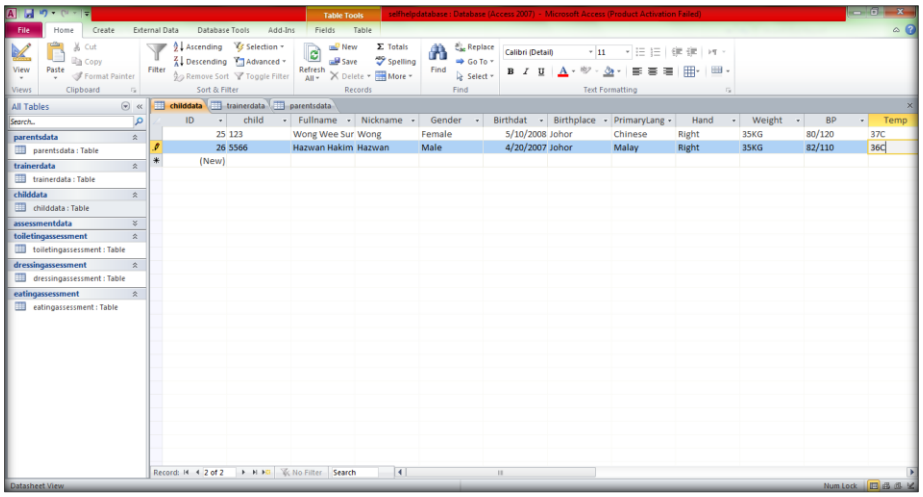
Through the user acceptance testing, it can be concluded that a user friendly system is preferable to the user. First of all, users are more attracted to use the system with interesting layout. Interesting layout can capture their interest to do perform assessment and training to the child. Next, the users put comfortable as one of their priority when using the system. Words with bigger font make them easier to read the instruction for assessment and training and audio implementation can make them

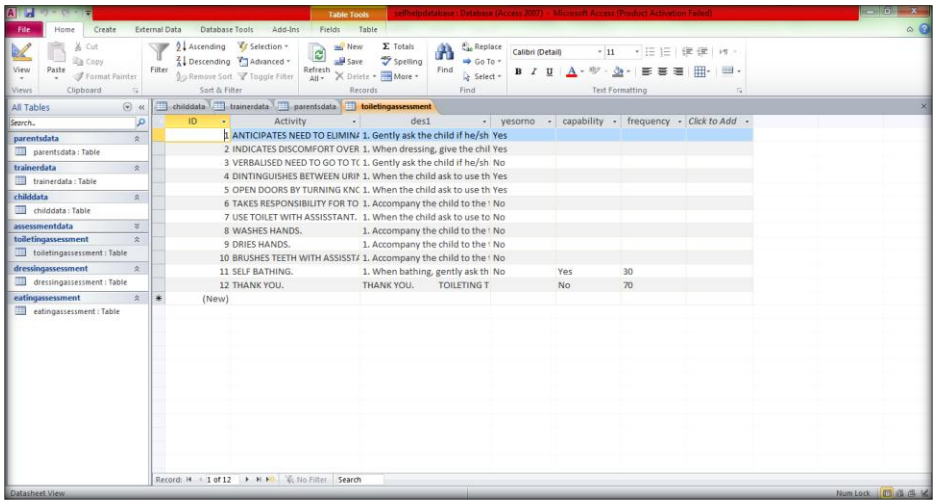
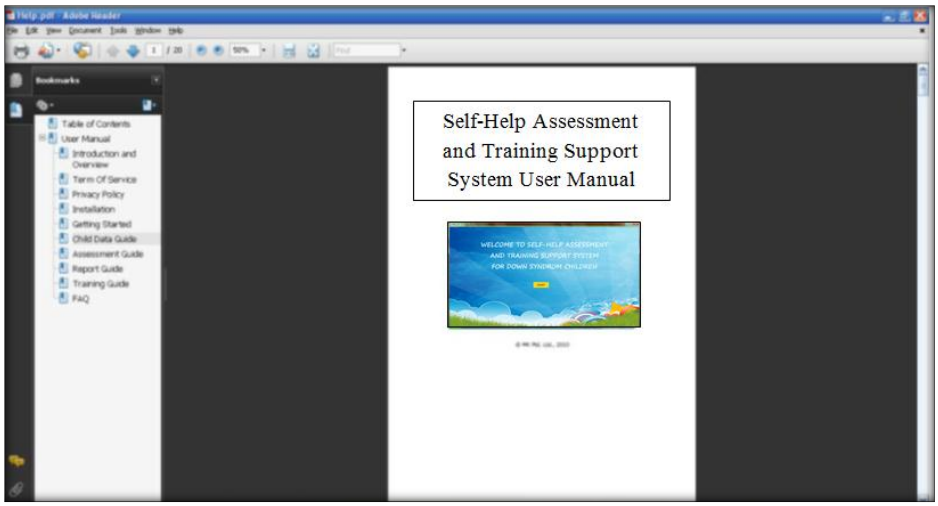
understand better on what to do when using the support system. However, the support system has some limitations. The software need to be close each time after assessment in order to generate the updated specified report, this can be due to the weakness in the coding to refresh instantly after each assessment is done. For further improvement, a solution to solve this matter is highly advisable. A research on new coding for the report that can refresh instantly after each assessment can be approached.

## 5.8 User Data Management

Table 5.6 shows how the children data and self-help database are manage and save within the self-help assessment and training support system.

**Table 5.11**: User data management

| Data management | Results |
|---|---|
| Patient data |  **Figure 5.11** Patient data<br><br>Microsoft Access is use as the medium to store all the patient data input |

| | |
|---|---|
| | which includes child data, parents data and trainer data. |
| Self-help database for assessment and training |  <br><br>**Figure 5.12** Self-help database<br><br>Microsoft Access is used to store all the self-help skills database for assessment and training. |
| Information on Down Syndrome and User manual |  <br><br><br>**Figure 5.13** Self-help assessment and training support system user manual<br><br>Information on Down Syndrome and the self-help assessment and training support system user manual is documented in PDF. |

# CHAPTER 6

# CONCLUSION AND RECOMMENDATION

## 6.1 Conclusion

Self-help database had been developed and classified into four sub-groups which are eating skills, dressing skills, toileting skills and sleeping skills. The database was carefully selected from international recognised curriculum for special children. Self-help assessment and training support system had been developed to evaluate and improve the self-help skills for the special children. Self-help support system is developed using C# programming language. The support system is useful for parents and trainers to do monitoring on the child's progress based on the assessment on self-help skills. Water level detection hardware had been implemented on the support system to help the assessment on eating ability specifically the ability to pour liquid into the cup. The hardware is to help the trainers in evaluating the child on the specified action.

## 6.2 Novelty

There are some novelties in this project. The novelties are:

- A whole new early intervention program software support system interface
- A Special features implemented such as voice instruction when operating the system for easier operational experience

- New method on generating report which generate pie chart that shows the assessment in percentage.
- New application on self-help assessment and training for special children.

## 6.3 Important result

There are several important results about the self-help assessment and training support system based on the testing. The important results are:

- The self-help assessment and training support system is entirely safe to use product.
- The database used for assessment and training is highly reliable.
- The result of the assessment can be used for data management centre due to its quantitative measurement features which generate data with percentage.
- The water that is use for the assessment and training are mainly for assessment and training purpose only and not for drinking.

## 6.4 Comparison with other methods

Other early intervention method such as the Orton Gullingham multisensory method and KinderKare training centre mainly uses trainer's experience on performing assessment and training over the child. There are no scientific records of successful rate on the training that given to the child. So it is hardly to say that the child had made a positive progress over the program. This is different with self-help assessment and training support system. This support system has a quantitative measurement over the assessment and training given to the child. A report is generate at the end of the assessment to shows the child's progress in percentage. An increment of percentage indicates that the child has a positive progress over the activity.

Other methods needed experience trainers to perform the program. Untrained parents need to send their children to the training centre to undergo assessment and training for at least once per week. This is ineffective due to its uncontinuous assessment. However, self-help assessment and training support system offers an easy to understand module for the parents to do assessment and training to their child. It can easily be installed inside a computer and instantly start the assessment. The support system has a database integrated inside the system which can help the parents to perform necessary assessment and training to their child by their own anytime they want.

## 6.5 Limitation of solution

Self-help assessment and training support system is highly reliable and effective for doing assessment and training on the child. However, there is some limitation in this project. One of the limitations is that the support system cannot generate the updated report after finish doing the assessment. User need to close the application and open it again so that the report can generate the updated assessment. This is may be due to the weakness on the coding. However, a study can be done to improve this weakness.

## 6.6 Recommendation

In the future, it is recommended that the assessment and training scope is expended for child more than 6 years old. This is to ensure the self-help assessment and training is continuous. The support system would be beneficial if the assessment and training activity is more interactive with the aid of visual or image. This implementation will give more clear instruction in executing the training activities.

# REFERENCES


[1] Peter Crosta M.A. What is Down syndrome?. 2010.

[2] Len Leshin, MD, FAAP. Trisomy 21: The Story of Down Syndrome. 2003.

[3] Encyclopedia of Childhood and Adolescence. Down Syndrome. 2001.

[4] Arthur Schoenstadt, MD. Down Syndrome. 2008.

[5] National Down syndrome society. Early Intervention. 2010.

[6] Peter W. D. Wright and Pamela Darr Wright. Early Intervention (Part C of IDEA). 2010.

[7] Buckley, S.J., & Sacks, B. *An overview of development of infants with Down syndrome (0-5 years)*. Portsmouth, UK: Down Syndrome Education International. 2001.

[8] Yap E.H. Cognitive assessment and training support system for special children. 2010.

[9] Buckley SJ, Bird G, Sacks B. Social development for individuals with Down syndrome - An overview. *Down Syndrome Issues and Information*. 2002.

[10] Ann Wheeler. Self Help Skills. 2009.

[11] Charles I. Scott, MD. What is Down Syndrome?. 2008.

[12] Tammi Reynolds, BA & Mark Dombeck, Ph.D. Genetic Causes of Mental Retardation - Down Syndrome. 2006.

[13] Tan M.K. development of interactive support system for language assessment and training for special children. 2010.

[14] Terri Mauro. What Are "Special Needs"?

[15] Yeo K.J., Eko Supriyanto, Haikal Satria, Tan M.K., Yap E.H.A Novel and Ubiquitous Early Intervention Support System for Down syndrome Children. 2010.

[16] Social Policy Research Centre. Evaluation plan: Early Intervention Program. 2006.

[17] New York State, Department of Health, Division of Family Health, Bureau of Early Intervention. State performance plan for the NYS early intervention program FFY 2005 – 2010. 2010.

[18] Liebovich B.T. Children's self-assessment.

[19] McCauley L. The development assessment of child development.

[20] Hills, T. W. Reaching potentials through appropriate assessment. In S. Bredekamp & T. Rosegrant (Eds.), *Reaching potentials: Appropriate curriculum and assessment for young children* (Vol. 1, pp. 43-63). Washington, DC: National Association for the Education of Young Children. (ERIC Document No. ED352160). 1992.

[21] Child & Adolescent Services Research Centre (CASRC). What is child development and what skills do children develop at different ages. 2008.

[22] Betterkiscare.psu.edu. All by myself: self-help skills in child care.

[23] ECMA international. C# programming language. 2006

[24] Utley C. Why should you move to C#. 2002.

[25] M. T. Schultheis, A. A. Razoo. The application of virtual technology in rehabilitation. 2001

[26] Susan barton. what is dyslexia? .Bright Solutions for Dyslexia, Inc. 2001

# APPENDIX A

## C Sharp Source Code for Self-Help Assessment and Training Support System

## Main Page

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace SelfHelpGUI
{
    public partial class MainForm : Form
    {
        public MainForm()
        {
            InitializeComponent();
        }

        private void start_Click(object sender, EventArgs e)
        {
            welcomeheader.Visible = false;
            mainheader.Visible = true;
            start.Visible = false;
            patientdatainput.Visible = true;
            assessment.Visible = true;
            report.Visible = true;
            training.Visible = true;

            System.Media.SoundPlayer sp = new
System.Media.SoundPlayer("C:\\Users\\Milton\\Documents\\Visual Studio
2010\\Projects\\SelfHelpGUI\\SelfHelpGUI\\Sound\\welcome2.wav");
            sp.Play();
        }

        private void patientdatainput_Click(object sender, EventArgs e)
        {
            patientdatainputform patientdatainputform = new patientdatainputform();
            patientdatainputform.Show();

        }

        private void assessment_Click(object sender, EventArgs e)
        {
            toiletinglabel assessmentform = new toiletinglabel();
            assessmentform.Show();
        }

        private void report_Click(object sender, EventArgs e)
        {
            reportform reportform = new reportform();
            reportform.Show();
        }

        private void training_Click(object sender, EventArgs e)
        {
```

```
            trainingform trainingform = new trainingform();
             trainingform.Show();
        }

        private void MainForm_Load(object sender, EventArgs e)
        {
            System.Media.SoundPlayer sp = new
System.Media.SoundPlayer("C:\\Users\\Milton\\Documents\\Visual Studio
2010\\Projects\\SelfHelpGUI\\SelfHelpGUI\\Sound\\welcome.wav");
            sp.Play();
        }
    }
}
```

## Patient Data Input

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace SelfHelpGUI
{
    public partial class patientdatainputform : Form
    {
        public patientdatainputform()
        {
            InitializeComponent();
        }

        private void trainerDI_Click(object sender, EventArgs e)
        {
            trainerdatainputform trainerdatainputform = new trainerdatainputform();
            trainerdatainputform.Show();
        }

        private void parentsDI_Click(object sender, EventArgs e)
        {
            parentsdatainputform parentsdatainputform = new parentsdatainputform();
            parentsdatainputform.Show();
        }

        private void childDI_Click_1(object sender, EventArgs e)
        {
            childdataform childdataform = new childdataform();
            childdataform.Show();
        }

        private void comboBox2_SelectedIndexChanged(object sender, EventArgs e)
        {
```

```
        }

        private void patientdatainputform_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the
'selfhelpdatabaseDataSet.childdata' table. You can move, or remove it, as needed.
            this.childdataTableAdapter.Fill(this.selfhelpdatabaseDataSet.childdata);

            System.Media.SoundPlayer sp = new
System.Media.SoundPlayer("C:\\Users\\Milton\\Documents\\Visual Studio
2010\\Projects\\SelfHelpGUI\\SelfHelpGUI\\Sound\\patientdatainput.wav");
            //sp.PlayLooping();
            sp.Play();

        }


    }
}
```

## Child Data Input

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.OleDb;

namespace SelfHelpGUI
{
    public partial class childdataform : Form
    {
        private OleDbConnection database;
        public childdataform()
        {
            InitializeComponent();
            database = new OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=C:\\Users\\Milton\\Documents\\Visual Studio
2010\\Projects\\SelfHelpGUI\\SelfHelpGUI\\selfhelpdatabase.accdb");
        }

        private void childdataform_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the
'selfhelpdatabaseDataSet.childdata' table. You can move, or remove it, as needed.
            this.childdataTableAdapter.Fill(this.selfhelpdatabaseDataSet.childdata);
        }
```

```
        private void CDsave_Click(object sender, EventArgs e)
        {
            OleDbCommand cmd = new OleDbCommand();
            cmd.CommandType = CommandType.Text;
            cmd.CommandText = "INSERT into
childdata(Fullname,Nickname,Gender,Birthdat,Birthplace,PrimaryLanguage,Hand,Weight,BP,Tem
p,HR,Allergy,Disease,Surgical,Hospital,Medicine,child) Values('"+

CDfullname.Text+"','"+CDnickname.Text+"','"+(string)CDgender.SelectedItem+"','"+CDdate.Va
lue+"','"+CDplace.Text+"','"+(string)CDlanguage.SelectedItem+"','"+

(string)CDhand.SelectedItem+"','"+CDweight.Text+"','"+CDBP.Text+"','"+CDtemp.Text+"','"+C
DHR.Text+"','"+CDallergy.Text+"','"+CDdisease.Text
+"','"+CDsurgical.Text+"','"+CDhospital.Text
                +"','"+CDmedicine.Text+"','"+CDid.Text+"')";
            cmd.Connection = database;
            database.Open();
            cmd.ExecuteNonQuery();
            database.Close();

            MessageBox.Show("Data Saved.", "Save",
            MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        }

        private void delete_Click(object sender, EventArgs e)
        {
            OleDbCommand cmd = new OleDbCommand();
            cmd.CommandType = CommandType.Text;
            cmd.CommandText = "delete from childdata where child ='" +CDid.Text+ "'";
            cmd.Connection = database;
            database.Open();
            cmd.ExecuteNonQuery();
            database.Close();

            MessageBox.Show("Data delete successful.", "Delete",
            MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        }
    }
}
```

## Parents Data Input

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.OleDb;

namespace SelfHelpGUI
{
```

```
    public partial class parentsdatainputform : Form
    {
        private OleDbConnection database;
        public parentsdatainputform()
        {
            InitializeComponent();
            database = new OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=C:\\Users\\Milton\\documents\\visual studio
2010\\Projects\\SelfHelpGUI\\SelfHelpGUI\\selfhelpdatabase.accdb");
        }

        private void parentsdatainputform_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the
'selfhelpdatabaseDataSet.parentsdata' table. You can move, or remove it, as needed.
            this.parentsdataTableAdapter.Fill(this.selfhelpdatabaseDataSet.parentsdata);
        }

        private void parentsDIsave_Click(object sender, EventArgs e)
        {
            OleDbCommand cmd = new OleDbCommand();
            cmd.CommandType = CommandType.Text;
            cmd.CommandText = ("insert into parentsdata
(ChildID,Fullname,birthdate,Birthplace,job,Workplace,Address,HP,Allergy,Disease,Surgical,
Hospital,Medicine) Values('"+childid2.Text+"','"+

PDname.Text+"','"+PDdate.Value+"','"+PDplace.Text+"','"+PDjob.Text+"','"+PDworkplace.Text
+"','"+PDaddress.Text+"','"+PDhp.Text+"','"+PDallergy.Text+"','"+PDdisease.Text+"','"+
                PDsurgical.Text+"','"+PDhospital.Text+"','"+PDmedicine.Text+"')");
            cmd.Connection = database;
            database.Open();
            cmd.ExecuteNonQuery();
            database.Close();

            MessageBox.Show("Data Saved.", "Save",
                MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        }

        private void delete_Click(object sender, EventArgs e)
        {
            OleDbCommand cmd = new OleDbCommand();
            cmd.CommandType = CommandType.Text;
            cmd.CommandText = "delete from parentsdata where ChildID='" + childid2.Text +
"'";
            cmd.Connection = database;
            database.Open();
            cmd.ExecuteNonQuery();
            database.Close();

            MessageBox.Show("Data delete successful.", "Delete",
            MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        }

        private void childid2_SelectedIndexChanged(object sender, EventArgs e)
        {

        }
    }
```

```
}
```

## Trainer Data Input

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.OleDb;

namespace SelfHelpGUI
{
    public partial class trainerdatainputform : Form
    {
        private OleDbConnection database;
        public trainerdatainputform()
        {
            InitializeComponent();
            database = new OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=C:\\Users\\Milton\\Documents\\Visual Studio
2010\\Projects\\SelfHelpGUI\\SelfHelpGUI\\selfhelpdatabase.accdb");
        }

        private void trainerdatainputform_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the
'selfhelpdatabaseDataSet.trainerdata' table. You can move, or remove it, as needed.
            this.trainerdataTableAdapter.Fill(this.selfhelpdatabaseDataSet.trainerdata);
        }

        private void TDsave_Click_1(object sender, EventArgs e)
        {


            OleDbCommand cmd = new OleDbCommand();
            cmd.CommandType = CommandType.Text;
            cmd.CommandText = "INSERT into
trainerdata(ChildID,Fullname,Birthdate,Birthplace,Job,Workplace,Address,HP)
Values('"+childid3.Text+"','" + TDname.Text + "','" +
                TDdate.Value + "','" + TDplace.Text + "','" + TDjob.Text + "','" +
TDworkplace.Text + "','" + TDaddress.Text + "','" + TDhp.Text + "')";
            cmd.Connection = database;
            database.Open();
            cmd.ExecuteNonQuery();
            database.Close();

            MessageBox.Show("Data Saved.", "Save",
            MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        }
```

```
        private void delete_Click(object sender, EventArgs e)
        {
            OleDbCommand cmd = new OleDbCommand();
            cmd.CommandType = CommandType.Text;
            cmd.CommandText = "delete from trainerdata where ChildID='" + childid3.Text +
"'";
            cmd.Connection = database;
            database.Open();
            cmd.ExecuteNonQuery();
            database.Close();

            MessageBox.Show("Data delete successful.", "Delete",
            MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        }
    }
}
```

## Assessment

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.OleDb;
using System.IO;
using System.Xml;
using System.IO.Ports;
using System.Threading;

namespace SelfHelpGUI
{
    public partial class toiletinglabel : Form
    {
        private int i1;
        private int i2;
        private int counter_yes = 0;
        private int counter_no = 0;
        private OleDbConnection database;
        private static string portNum = "COM4";

        public toiletinglabel()
        {
            InitializeComponent();
            RxString = "";
            database = new OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=C:\\Users\\Milton\\Documents\\Visual Studio
2010\\Projects\\SelfHelpGUI\\SelfHelpGUI\\selfhelpdatabase.accdb");
        }
```

```
        private void assessmentform_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the
'selfhelpdatabaseDataSet.eatingassessment' table. You can move, or remove it, as needed.

this.eatingassessmentTableAdapter.Fill(this.selfhelpdatabaseDataSet.eatingassessment);
            // TODO: This line of code loads data into the
'selfhelpdatabaseDataSet.dressingassessment' table. You can move, or remove it, as needed.

this.dressingassessmentTableAdapter.Fill(this.selfhelpdatabaseDataSet.dressingassessment);
            // TODO: This line of code loads data into the
'selfhelpdatabaseDataSet.toiletingassessment' table. You can move, or remove it, as
needed.

this.toiletingassessmentTableAdapter.Fill(this.selfhelpdatabaseDataSet.toiletingassessmen
t);
            // TODO: This line of code loads data into the
'selfhelpdatabaseDataSet.assessmentdata' table. You can move, or remove it, as needed.

this.assessmentdataTableAdapter.Fill(this.selfhelpdatabaseDataSet.assessmentdata);

            System.Media.SoundPlayer sp = new
System.Media.SoundPlayer("C:\\Users\\Milton\\Documents\\Visual Studio
2010\\Projects\\SelfHelpGUI\\SelfHelpGUI\\Sound\\assessment.wav");
            sp.Play();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            pictureBox2.Visible = false;
            pictureBox3.Visible = false;
            ir1.Visible = false;
            ir2.Visible = false;
            i2 = counter_no;
            i1 = counter_yes++;


            if (sleepingrb.Checked)
            {
                sleepingyesorno.Text = "Yes";
                OleDbCommand cmd = new OleDbCommand();
                cmd.CommandType = CommandType.Text;
                cmd.CommandText = "update assessmentdata set yesorno='" +
sleepingyesorno.Text + "' where (Activity='" + sleepingdescription.Text + "')";
                cmd.Connection = database;
                database.Open();
                cmd.ExecuteNonQuery();
                database.Close();
                this.assessmentdataBindingSource.MoveNext();

                if (i1 + i2 > 7)
                {
                    float total = i1 + i2;
                    float deg1 = (i1 / total) * 100;
                    float deg2 = (i2 / total) * 100;
                    float deg11 = 5 * (int)Math.Round(deg1 / 5.15);
```

```
                    sleepingfrequency.Text = deg11.ToString();

                    cmd.CommandText = "update assessmentdata set frequency='" +
sleepingfrequency.Text + "' where (Activity='" + sleepingdescription.Text + "')";
                    cmd.Connection = database;
                    database.Open();
                    cmd.ExecuteNonQuery();
                    database.Close();
                }

                if (i1 + i2 > 8)
                {
                    float total = i1 + i2;
                    float deg1 = (i1 / total) * 100;
                    float deg2 = (i2 / total) * 100;
                    float deg22 = 5 * (int)Math.Round(deg2 / 5.15);
                    sleepingfrequency.Text = deg22.ToString();

                    cmd.CommandText = "update assessmentdata set frequency='" +
sleepingfrequency.Text + "' where (Activity='" + sleepingdescription.Text + "')";
                    cmd.Connection = database;
                    database.Open();
                    cmd.ExecuteNonQuery();
                    database.Close();

                    MessageBox.Show("You have finish the assessment. You can check the
result at the 'Report' section.", "Well Done",
                    MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
                    assessmentno.Visible = false;
                    assessmentyes.Visible = false;
                    assessmentask.Visible = false;
                    ask2.Visible = true;
                }
            }

            if (toiletingrb.Checked)
            {
                toiletingyesorno.Text = "Yes";
                OleDbCommand cmd = new OleDbCommand();
                cmd.CommandType = CommandType.Text;
                cmd.CommandText = "update toiletingassessment set yesorno='" +
toiletingyesorno.Text + "' where (Activity='" + toiletingdescription.Text + "')";
                cmd.Connection = database;
                database.Open();
                cmd.ExecuteNonQuery();
                database.Close();
                this.toiletingassessmentBindingSource.MoveNext();

                if (i1 + i2 > 8)
                {
                    float total = i1 + i2;
                    float deg1 = (i1 / total) * 100;
                    float deg2 = (i2 / total) * 100;
                    float deg11 = 5* (int)Math.Round(deg1 / 5.15);
                    toiletingfrequency.Text = deg11.ToString();

                    cmd.CommandText = "update toiletingassessment set frequency='" +
toiletingfrequency.Text + "' where (Activity='" + toiletingdescription.Text + "')";
```

```
                    cmd.Connection = database;
                    database.Open();
                    cmd.ExecuteNonQuery();
                    database.Close();
                }

                if (i1 + i2 > 9)
                {
                    float total = i1 + i2;
                    float deg1 = (i1 / total) * 100;
                    float deg2 = (i2 / total) * 100;
                    float deg22 = 5* (int)Math.Round(deg2 / 5.15);
                    toiletingfrequency.Text = deg22.ToString();

                    cmd.CommandText = "update toiletingassessment set frequency='" +
toiletingfrequency.Text + "' where (Activity='" + toiletingdescription.Text + "')";
                    cmd.Connection = database;
                    database.Open();
                    cmd.ExecuteNonQuery();
                    database.Close();

                    MessageBox.Show("You have finish the assessment. You can check the
result at the 'Report' section.", "Well Done",
                    MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
                    assessmentno.Visible = false;
                    assessmentyes.Visible = false;
                    assessmentask.Visible = false;
                    ask2.Visible = true;
                }
            }

            if (dressingrb.Checked)
            {

                dressingyesorno.Text = "Yes";
                OleDbCommand cmd = new OleDbCommand();
                cmd.CommandType = CommandType.Text;
                cmd.CommandText = "update dressingassessment set yesorno='" +
dressingyesorno.Text + "' where (Activity='" + dressingdescription.Text + "')";
                cmd.Connection = database;
                database.Open();
                cmd.ExecuteNonQuery();
                database.Close();
                this.dressingassessmentBindingSource.MoveNext();

                if (i1 + i2 > 12)
                {
                    float total = i1 + i2;
                    float deg1 = (i1 / total) * 100;
                    float deg2 = (i2 / total) * 100;
                    float deg11 = 5 * (int)Math.Round(deg1 / 5.15);
                    dressingfrequency.Text = deg11.ToString();

                    cmd.CommandText = "update dressingassessment set frequency='" +
dressingfrequency.Text + "' where (Activity='" + dressingdescription.Text + "')";
                    cmd.Connection = database;
                    database.Open();
                    cmd.ExecuteNonQuery();
```

```
                    database.Close();
                }

                if (i1 + i2 > 13)
                {
                    float total = i1 + i2;
                    float deg1 = (i1 / total) * 100;
                    float deg2 = (i2 / total) * 100;
                    float deg22 = 5 * (int)Math.Round(deg2 / 5.15);
                    dressingfrequency.Text = deg22.ToString();

                    cmd.CommandText = "update dressingassessment set frequency='" +
dressingfrequency.Text + "' where (Activity='" + dressingdescription.Text + "')";
                    cmd.Connection = database;
                    database.Open();
                    cmd.ExecuteNonQuery();
                    database.Close();

                    MessageBox.Show("You have finish the assessment. You can check the
result at the 'Report' section.", "Well Done",
                    MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
                    assessmentno.Visible = false;
                    assessmentyes.Visible = false;
                    assessmentask.Visible = false;
                    ask2.Visible = true;
                }
            }

            if (eatingrb.Checked)
            {
                eatingyesorno.Text = "Yes";
                OleDbCommand cmd = new OleDbCommand();
                cmd.CommandType = CommandType.Text;
                cmd.CommandText = "update eatingassessment set yesorno='" +
eatingyesorno.Text + "' where (Activity='" + eatingdescription.Text + "')";
                cmd.Connection = database;
                database.Open();
                cmd.ExecuteNonQuery();
                database.Close();
                this.eatingassessmentBindingSource.MoveNext();

                if (i1 + i2 > 32)
                {
                    float total = i1 + i2;
                    float deg1 = (i1 / total) * 100;
                    float deg2 = (i2 / total) * 100;
                    float deg11 = 5 * (int)Math.Round(deg1 / 5.15);
                    eatingfrequency.Text = deg11.ToString();

                    cmd.CommandText = "update eatingassessment set frequency='" +
eatingfrequency.Text + "' where (Activity='" + eatingdescription.Text + "')";
                    cmd.Connection = database;
                    database.Open();
                    cmd.ExecuteNonQuery();
                    database.Close();
                }

                if (i1 + i2 > 33)
```

```
                {
                    float total = i1 + i2;
                    float deg1 = (i1 / total) * 100;
                    float deg2 = (i2 / total) * 100;
                    float deg22 = 5 * (int)Math.Round(deg2 / 5.15);
                    eatingfrequency.Text = deg22.ToString();

                    cmd.CommandText = "update eatingassessment set frequency='" +
eatingfrequency.Text + "' where (Activity='" + eatingdescription.Text + "')";
                    cmd.Connection = database;
                    database.Open();
                    cmd.ExecuteNonQuery();
                    database.Close();

                    MessageBox.Show("You have finish the assessment. You can check the
result at the 'Report' section.", "Well Done",
                    MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
                    assessmentno.Visible = false;
                    assessmentyes.Visible = false;
                    assessmentask.Visible = false;
                    ask2.Visible = true;
                }

            }
    }

        private void assessmentno_Click(object sender, EventArgs e)
        {
            pictureBox2.Visible = false;
            pictureBox3.Visible = false;
            ir1.Visible = false;
            ir2.Visible = false;
            i2 = counter_no++;
            i1 = counter_yes;


            if (sleepingrb.Checked)
            {
                sleepingyesorno.Text = "No";
                OleDbCommand cmd = new OleDbCommand();
                cmd.CommandType = CommandType.Text;
                cmd.CommandText = "update assessmentdata set yesorno='" +
sleepingyesorno.Text + "' where (Activity='" + sleepingdescription.Text + "')";
                cmd.Connection = database;
                database.Open();
                cmd.ExecuteNonQuery();
                database.Close();
                this.assessmentdataBindingSource.MoveNext();

                if (i1 + i2 > 7)
                {
                    float total = i1 + i2;
                    float deg1 = (i1 / total) * 100;
                    float deg2 = (i2 / total) * 100;
                    float deg11 = 5 * (int)Math.Round(deg1 / 5.15);
                    sleepingfrequency.Text = deg11.ToString();
```

```
                    cmd.CommandText = "update assessmentdata set frequency='" +
sleepingfrequency.Text + "' where (Activity='" + sleepingdescription.Text + "')";
                    cmd.Connection = database;
                    database.Open();
                    cmd.ExecuteNonQuery();
                    database.Close();
                }

                 if (i1 + i2 > 8)
                {
                    float total = i1 + i2;
                    float deg1 = (i1 / total) * 100;
                    float deg2 = (i2 / total) * 100;
                    float deg22 = 5 * (int)Math.Round(deg2 / 5.15);
                    sleepingfrequency.Text = deg22.ToString();

                    cmd.CommandText = "update assessmentdata set frequency='" +
sleepingfrequency.Text + "' where (Activity='" + sleepingdescription.Text + "')";
                    cmd.Connection = database;
                    database.Open();
                    cmd.ExecuteNonQuery();
                    database.Close();

                    MessageBox.Show("You have finish the assessment. You can check the
result at the 'Report' section.", "Well Done",
                    MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
                    assessmentno.Visible = false;
                    assessmentyes.Visible = false;
                    assessmentask.Visible = false;
                    ask2.Visible = true;
                }
            }

            if (toiletingrb.Checked)
            {
                toiletingyesorno.Text = "No";
                OleDbCommand cmd = new OleDbCommand();
                cmd.CommandType = CommandType.Text;
                cmd.CommandText = "update toiletingassessment set yesorno='" +
toiletingyesorno.Text + "' where (Activity='" + toiletingdescription.Text + "')";
                cmd.Connection = database;
                database.Open();
                cmd.ExecuteNonQuery();
                database.Close();
                this.toiletingassessmentBindingSource.MoveNext();

                if (i1 + i2 > 8)
                {
                    float total = i1 + i2;
                    float deg1 = (i1 / total) * 100;
                    float deg2 = (i2 / total) * 100;
                    float deg11 = 5* (int)Math.Round(deg1 / 5.15);
                    toiletingfrequency.Text = deg11.ToString();

                    cmd.CommandText = "update toiletingassessment set frequency='" +
toiletingfrequency.Text + "' where (Activity='" + toiletingdescription.Text + "')";
                    cmd.Connection = database;
                    database.Open();
```

```
                    cmd.ExecuteNonQuery();
                    database.Close();
                }

                if (i1 +i2 > 9)
                {
                    float total = i1 + i2;
                    float deg1 = (i1 / total) * 100;
                    float deg2 = (i2 / total) * 100;
                    float deg22 = 5* (int)Math.Round(deg2 / 5.15);
                    toiletingfrequency.Text = deg22.ToString();

                    cmd.CommandText = "update toiletingassessment set frequency='" +
toiletingfrequency.Text + "' where (Activity='" + toiletingdescription.Text + "')";
                    cmd.Connection = database;
                    database.Open();
                    cmd.ExecuteNonQuery();
                    database.Close();
                    MessageBox.Show("You have finish the assessment. You can check the
result at the 'Report' section.", "Well Done",
                    MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
                    assessmentno.Visible = false;
                    assessmentyes.Visible = false;
                    assessmentask.Visible = false;
                    ask2.Visible = true;
                }



            }

            if (dressingrb.Checked)
            {
                dressingyesorno.Text = "No";
                OleDbCommand cmd = new OleDbCommand();
                cmd.CommandType = CommandType.Text;
                cmd.CommandText = "update dressingassessment set yesorno='" +
dressingyesorno.Text + "' where (Activity='" + dressingdescription.Text + "')";
                cmd.Connection = database;
                database.Open();
                cmd.ExecuteNonQuery();
                database.Close();
                this.dressingassessmentBindingSource.MoveNext();

                if (i1 + i2 > 12)
                {
                    float total = i1 + i2;
                    float deg1 = (i1 / total) * 100;
                    float deg2 = (i2 / total) * 100;
                    float deg11 = 5 * (int)Math.Round(deg1 / 5.15);
                    dressingfrequency.Text = deg11.ToString();

                    cmd.CommandText = "update dressingassessment set frequency='" +
dressingfrequency.Text + "' where (Activity='" + dressingdescription.Text + "')";
                    cmd.Connection = database;
                    database.Open();
                    cmd.ExecuteNonQuery();
                    database.Close();
```

```
                }

                if (i1 + i2 > 13)
                {
                    float total = i1 + i2;
                    float deg1 = (i1 / total) * 100;
                    float deg2 = (i2 / total) * 100;
                    float deg22 = 5 * (int)Math.Round(deg2 / 5.15);
                    dressingfrequency.Text = deg22.ToString();

                    cmd.CommandText = "update dressingassessment set frequency='" +
dressingfrequency.Text + "' where (Activity='" + dressingdescription.Text + "')";
                    cmd.Connection = database;
                    database.Open();
                    cmd.ExecuteNonQuery();
                    database.Close();

                    MessageBox.Show("You have finish the assessment. You can check the
result at the 'Report' section.", "Well Done",
                    MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
                    assessmentno.Visible = false;
                    assessmentyes.Visible = false;
                    assessmentask.Visible = false;
                    ask2.Visible = true;
                }
            }

            if (eatingrb.Checked)
            {
                eatingyesorno.Text = "No";
                OleDbCommand cmd = new OleDbCommand();
                cmd.CommandType = CommandType.Text;
                cmd.CommandText = "update eatingassessment set yesorno='" +
eatingyesorno.Text + "' where (Activity='" + eatingdescription.Text + "')";
                cmd.Connection = database;
                database.Open();
                cmd.ExecuteNonQuery();
                database.Close();
                this.eatingassessmentBindingSource.MoveNext();

               if (i1 + i2 > 32)
                {
                    float total = i1 + i2;
                    float deg1 = (i1 / total) * 100;
                    float deg2 = (i2 / total) * 100;
                    float deg11 = 5 * (int)Math.Round(deg1 / 5.15);
                    eatingfrequency.Text = deg11.ToString();

                    cmd.CommandText = "update eatingassessment set frequency='" +
eatingfrequency.Text + "' where (Activity='" + eatingdescription.Text + "')";
                    cmd.Connection = database;
                    database.Open();
                    cmd.ExecuteNonQuery();
                    database.Close();
                }

                if (i1 + i2 > 33)
                {
```

```
                    float total = i1 + i2;
                    float deg1 = (i1 / total) * 100;
                    float deg2 = (i2 / total) * 100;
                    float deg22 = 5 * (int)Math.Round(deg2 / 5.15);
                    eatingfrequency.Text = deg22.ToString();

                    cmd.CommandText = "update eatingassessment set frequency='" +
eatingfrequency.Text + "' where (Activity='" + eatingdescription.Text + "')";
                    cmd.Connection = database;
                    database.Open();
                    cmd.ExecuteNonQuery();
                    database.Close();

                    MessageBox.Show("You have finish the assessment. You can check the
result at the 'Report' section.", "Well Done",
                    MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
                    assessmentno.Visible = false;
                    assessmentyes.Visible = false;
                    assessmentask.Visible = false;
                    ask2.Visible = true;
                }

            }
        }

        private void dressingrb_CheckedChanged_1(object sender, EventArgs e)
        {
            pictureBox2.Visible = false;
            pictureBox3.Visible = false;
            ir1.Visible = false;
            ir2.Visible = false;

            counter_yes = 0;
            counter_no = 0;
            assessmentask.Visible = true;
            ask2.Visible = false;
            assessmentwelcome.Visible = false;
            assessmentask.Visible = true;
            assessmentno.Visible = true;
            assessmentyes.Visible = true;
            eatinglabel.Visible = false;
            dressinglabel.Visible = true;
            toiletinglabel1.Visible = false;
            sleepinglabel.Visible = false;
            eatingdescription.Visible = false;
            dressingdescription.Visible = true;
            toiletingdescription.Visible = false;
            sleepingdescription.Visible = false;
            this.dressingassessmentBindingSource.MoveFirst();
        }

        private void eatingrb_CheckedChanged(object sender, EventArgs e)
        {
            if (eatingrb.Checked)
            {
              //  mySerial = new SerialPort(portNum, 9600, Parity.None, 8, StopBits.One);
             //   mySerial.DataReceived += new
SerialDataReceivedEventHandler(this.ReadSerial);
```

```
             //   mySerial.Open();
             //   if (mySerial.IsOpen)
              //  eatingrb.Enabled = false;

                System.Media.SoundPlayer sp = new
System.Media.SoundPlayer("C:\\Users\\Milton\\Documents\\Visual Studio
2010\\Projects\\SelfHelpGUI\\SelfHelpGUI\\Sound\\eatingassessment2.wav");
                sp.Play();
            }

            ir1.Visible = true;
            ir2.Visible = true;
            pictureBox3.Visible = true;
            pictureBox2.Visible = true;
            counter_yes = 0;
            counter_no = 0;
            assessmentask.Visible = true;
            ask2.Visible = false;
            assessmentwelcome.Visible = false;
            assessmentask.Visible = true;
            assessmentno.Visible = true;
            assessmentyes.Visible = true;
            eatinglabel.Visible = true;
            dressinglabel.Visible = false;
            toiletinglabel1.Visible = false;
            sleepinglabel.Visible = false;
            eatingdescription.Visible = true;
            dressingdescription.Visible = false;
            toiletingdescription.Visible = false;
            sleepingdescription.Visible = false;
            this.eatingassessmentBindingSource.MoveFirst();
        }

        private void sleepingrb_CheckedChanged_1(object sender, EventArgs e)
        {
            pictureBox2.Visible = false;
            pictureBox3.Visible = false;
            ir1.Visible = false;
            ir2.Visible = false;

            counter_yes = 0;
            counter_no = 0;
            assessmentask.Visible = true;
            ask2.Visible = false;
            assessmentwelcome.Visible = false;
            assessmentask.Visible = true;
            assessmentno.Visible = true;
            assessmentyes.Visible = true;
            sleepinglabel.Visible = true;
            eatinglabel.Visible = false;
            dressinglabel.Visible = false;
            toiletinglabel1.Visible = false;
            sleepinglabel.Visible = true;
            dressingdescription.Visible = false;
            toiletingdescription.Visible = false;
            sleepingdescription.Visible = true;

            this.assessmentdataBindingSource.MoveFirst();
```

```
        }

        private void toiletingrb_CheckedChanged_1(object sender, EventArgs e)
        {
            pictureBox2.Visible = false;
            pictureBox3.Visible = false;
            ir1.Visible = false;
            ir2.Visible = false;

            counter_yes = 0;
            counter_no = 0;
            assessmentask.Visible = true;
            ask2.Visible = false;
            assessmentwelcome.Visible = false;
            assessmentask.Visible = true;
            assessmentno.Visible = true;
            assessmentyes.Visible = true;
            sleepinglabel.Visible = true;
            eatinglabel.Visible = false;
            dressinglabel.Visible = false;
            toiletinglabel1.Visible = true;
            sleepinglabel.Visible = false;
            dressingdescription.Visible = false;
            toiletingdescription.Visible = true;
            sleepingdescription.Visible = false;
            this.assessmentdataBindingSource.MoveFirst();
        }



        //for reading
        private static SerialPort mySerial;
        private String RxString;
        private String Reserve;

        public void ConnectSerial()
        {
            if (mySerial != null)
            {
                mySerial = new SerialPort(portNum, 115200, Parity.None, 8, StopBits.One);
                mySerial.DataReceived += new
SerialDataReceivedEventHandler(this.ReadSerial);
            }
            if (!mySerial.IsOpen)
                mySerial.Open();

        }



        public void ReadSerial(object sender, System.IO.Ports.SerialDataReceivedEventArgs
e)
        {
            try
            {
                while (true)
                {
                    byte[] mybyte = new byte[1024];
```

```
                    int intbyte = mySerial.Read(mybyte, 0, 1024);
                    if (intbyte == 0)
                        break;
                    int x = 0;
                    for (int i = 0; i < intbyte; i++)
                    {
                        x = Convert.ToInt32(mybyte[i]);
                        if (x == 170)
                            Reserve = x.ToString() + ";";
                        if (105 < x && x < 122)
                            Reserve += x.ToString() + ";";
                        if (123 < x && x < 143)
                            Reserve += x.ToString() + ";";
                        if (x == 187)
                        {
                            Reserve += x.ToString();
                            this.Invoke(new EventHandler(SerialDataCheck));

                        }
                    }

                }
            }
            catch (Exception ex)
            { }
        }

        private void SerialDataCheck(object sender, EventArgs e)
        {
            textBox2.AppendText(String.Format("{0} \r\n", Reserve));
            String[] xsplit = Reserve.Split(';');
            textBox4.Text = xsplit[2];
            if (((xsplit[2].Equals("105")) || (xsplit[2].Equals("106")) ||
(xsplit[2].Equals("107")) || (xsplit[2].Equals("108")) || (xsplit[2].Equals("109")) ||
(xsplit[2].Equals("110")) || (xsplit[2].Equals("111")) || (xsplit[2].Equals("112")) ||
(xsplit[2].Equals("113")) ||
                (xsplit[2].Equals("114")) || (xsplit[2].Equals("115")) ||
(xsplit[2].Equals("116")) || (xsplit[2].Equals("117")) || (xsplit[2].Equals("118")) ||
(xsplit[2].Equals("119")) || (xsplit[2].Equals("120")) ||
                (xsplit[2].Equals("121")) || (xsplit[2].Equals("122"))))
            {
                textBox3.Text = "FULL";
                pictureBox2.Visible = true;
                pictureBox3.Visible = false;
                ir1.BackColor = Color.Red;
                ir2.BackColor = Color.Green;
            }
            else if ((xsplit[1].Equals("123")) || (xsplit[1].Equals("124")) ||
(xsplit[1].Equals("125")) || (xsplit[1].Equals("126")) || (xsplit[1].Equals("127")) ||
(xsplit[1].Equals("128")) ||
                (xsplit[1].Equals("136")) || (xsplit[1].Equals("137")) ||
(xsplit[1].Equals("138")) || (xsplit[1].Equals("139")) || (xsplit[1].Equals("140")) ||
(xsplit[1].Equals("141")) || (xsplit[1].Equals("142")) ||
                (xsplit[1].Equals("129")) || (xsplit[1].Equals("130")) ||
(xsplit[1].Equals("131")) || (xsplit[1].Equals("132")) || (xsplit[1].Equals("133")) ||
(xsplit[1].Equals("134")) || (xsplit[1].Equals("135")) ||
```

```
                (xsplit[1].Equals("136")) || (xsplit[1].Equals("137")) ||
(xsplit[1].Equals("138")) || (xsplit[1].Equals("139")) || (xsplit[1].Equals("140")) ||
(xsplit[1].Equals("141")) || (xsplit[1].Equals("142")) || (xsplit[1].Equals("143")))
            {
                textBox3.Text = "HALF EMPTY";
                pictureBox2.Visible = false;
                pictureBox3.Visible = true;
                ir1.BackColor = Color.Red;
                ir2.BackColor = Color.Black;
            }

            else
            {
                textBox3.Text = "empty";
                pictureBox2.Visible = false;
                pictureBox3.Visible = true;
                ir1.BackColor = Color.Black;
                ir2.BackColor = Color.Black;
            }
            //UpdateGUI();

        }


        private void StopSerial()
        {
            if (mySerial.IsOpen)
            {
                mySerial.Close();
                mySerial = null;
                RxString = "";
                eatingrb.Enabled = true;
            }
        }

        private void ir1_Paint(object sender, PaintEventArgs e)
        {

        }

        private void ir2_Paint(object sender, PaintEventArgs e)
        {
            if (ir2.BackColor == Color.Green)
            {
                this.StopSerial();
                eatingrb.Enabled = true;
                System.Media.SoundPlayer sp = new
System.Media.SoundPlayer("C:\\Users\\Milton\\Documents\\Visual Studio
2010\\Projects\\SelfHelpGUI\\SelfHelpGUI\\Sound\\cupfull.wav");
                sp.Play();
            }
        }

    }
}
```

## Report

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Drawing.Imaging;

namespace SelfHelpGUI
{
    public partial class reportform : Form
    {
        public reportform()
        {
            InitializeComponent();
        }
        private void reportform_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the
'selfhelpdatabaseDataSet.eatingassessment' table. You can move, or remove it, as needed.

this.eatingassessmentTableAdapter.Fill(this.selfhelpdatabaseDataSet.eatingassessment);
            // TODO: This line of code loads data into the
'selfhelpdatabaseDataSet.dressingassessment' table. You can move, or remove it, as needed.

this.dressingassessmentTableAdapter.Fill(this.selfhelpdatabaseDataSet.dressingassessment);
            // TODO: This line of code loads data into the
'selfhelpdatabaseDataSet.assessmentdata' table. You can move, or remove it, as needed.

this.assessmentdataTableAdapter.Fill(this.selfhelpdatabaseDataSet.assessmentdata);
            // TODO: This line of code loads data into the
'selfhelpdatabaseDataSet.toiletingassessment' table. You can move, or remove it, as
needed.

this.toiletingassessmentTableAdapter.Fill(this.selfhelpdatabaseDataSet.toiletingassessmen
t);

            System.Media.SoundPlayer sp = new
System.Media.SoundPlayer("C:\\Users\\Milton\\Documents\\Visual Studio
2010\\Projects\\SelfHelpGUI\\SelfHelpGUI\\Sound\\report.wav");
            sp.Play();
        }

        private void eatingrb_CheckedChanged(object sender, EventArgs e)
        {

this.eatingassessmentTableAdapter.Fill(selfhelpdatabaseDataSet.eatingassessment);

            reportsave.Visible = true;
            reporttext.Visible = false;
            eatingchart.Visible = true;
            dressingchart.Visible = false;
            toiletingchart.Visible = false;
```

```
            sleepingchart.Visible = false;


        }

        private void toiletingrb_CheckedChanged(object sender, EventArgs e)
        {
            reportsave.Visible = true;
            reporttext.Visible = false;
            eatingchart.Visible = false;
            dressingchart.Visible = false;
            toiletingchart.Visible = true;
            sleepingchart.Visible = false;
        }

        private void dressingrb_CheckedChanged(object sender, EventArgs e)
        {
            reportsave.Visible = true;
            reporttext.Visible = false;
            eatingchart.Visible = false;
            dressingchart.Visible = true;
            toiletingchart.Visible = false;
            sleepingchart.Visible = false;
        }

        private void sleepingrb_CheckedChanged(object sender, EventArgs e)
        {
            reportsave.Visible = true;
            reporttext.Visible = false;
            eatingchart.Visible = false;
            dressingchart.Visible = false;
            toiletingchart.Visible = false;
            sleepingchart.Visible = true;
        }

        private void button1_Click(object sender, EventArgs e)
        {
            if (eatingrb.Checked)
            {
                eatingchart.SaveImage("C:\\Users\\Milton\\Desktop\\eatingreport.jpg",
ImageFormat.Jpeg);

                MessageBox.Show("Data Saved.", "Save",
                MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
            }

            if (dressingrb.Checked)
            {
                dressingchart.SaveImage("C:\\Users\\Milton\\Desktop\\dressingreport.jpg",
ImageFormat.Jpeg);

                MessageBox.Show("Data Saved.", "Save",
                MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
            }

            if (toiletingrb.Checked)
            {
```

```
toiletingchart.SaveImage("C:\\Users\\Milton\\Desktop\\toiletingreport.jpg",
System.Drawing.Imaging.ImageFormat.Jpeg);

                MessageBox.Show("Data Saved.", "Save",
                MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
            }

            if (sleepingrb.Checked)
            {
                sleepingchart.SaveImage("C:\\Users\\Milton\\Desktop\\sleepingreport.jpg",
System.Drawing.Imaging.ImageFormat.Jpeg);

                MessageBox.Show("Data Saved.", "Save",
                MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
            }

        }
    }
}
```

## Training

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.OleDb;

namespace SelfHelpGUI
{
    public partial class trainingform : Form
    {
        private OleDbConnection database;

        public trainingform()
        {
            InitializeComponent();
            database = new OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=C:\\Users\\Milton\\Documents\\Visual Studio
2010\\Projects\\SelfHelpGUI\\SelfHelpGUI\\selfhelpdatabase.accdb");
        }

        private void trainingform_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the
'selfhelpdatabaseDataSet1.eatingassessment' table. You can move, or remove it, as needed.

this.eatingassessmentTableAdapter.Fill(this.selfhelpdatabaseDataSet1.eatingassessment);
```

```
            // TODO: This line of code loads data into the
'selfhelpdatabaseDataSet.dressingassessment' table. You can move, or remove it, as needed.

this.dressingassessmentTableAdapter.Fill(this.selfhelpdatabaseDataSet.dressingassessment);
            // TODO: This line of code loads data into the
'selfhelpdatabaseDataSet.toiletingassessment' table. You can move, or remove it, as
needed.

this.toiletingassessmentTableAdapter.Fill(this.selfhelpdatabaseDataSet.toiletingassessmen
t);
            // TODO: This line of code loads data into the
'selfhelpdatabaseDataSet.assessmentdata' table. You can move, or remove it, as needed.

this.assessmentdataTableAdapter.Fill(this.selfhelpdatabaseDataSet.assessmentdata);

            System.Media.SoundPlayer sp = new
System.Media.SoundPlayer("C:\\Users\\Milton\\Documents\\Visual Studio
2010\\Projects\\SelfHelpGUI\\SelfHelpGUI\\Sound\\training.wav");
            //sp.PlayLooping();
            sp.Play();
        }

        private void trainingwelcome_Click(object sender, EventArgs e)
        {

        }

        private void assessmentyes_Click(object sender, EventArgs e)
        {
            if (eatingrb.Checked)
            {
                this.eatingassessmentBindingSource.MoveNext();
            }

            if (dressingrb.Checked)
            {
             this.dressingassessmentBindingSource.MoveNext();
            }

            if (toiletingrb.Checked)
            {
                this.toiletingassessmentBindingSource.MoveNext();
            }

            if (sleepingrb.Checked)
            {
                this.assessmentdataBindingSource.MoveNext();
            }
        }

        private void trainingback_Click(object sender, EventArgs e)
        {
            if (eatingrb.Checked)
            {
                this.eatingassessmentBindingSource.MovePrevious();
            }

            if (dressingrb.Checked)
```

```
    {
        this.dressingassessmentBindingSource.MovePrevious();
    }

    if (toiletingrb.Checked)
    {
     this.toiletingassessmentBindingSource.MovePrevious();

    }

   if (sleepingrb.Checked)
    {
        this.assessmentdataBindingSource.MovePrevious();

    }
}

private void sleepingrb_CheckedChanged(object sender, EventArgs e)
{
    trainingwelcome.Visible = false;
    trainingback.Visible = true;
    trainingnext.Visible = true;
    eatinglabel.Visible = false;
    dressinglabel.Visible = false;
    toiletinglabel1.Visible = false;
    sleepinglabel.Visible = true;
    eatingdescription.Visible = false;
    eatingactivity.Visible = false;
    dressingactivity.Visible = false;
    dressingdescription.Visible = false;
    toiletingdescription.Visible = false;
    toiletingactivity.Visible = false;
    sleepingdescription.Visible = true;
    sleepingactivity.Visible = true;
    this.assessmentdataBindingSource.MoveFirst();
}

private void eatingrb_CheckedChanged(object sender, EventArgs e)
{
    trainingwelcome.Visible = false;
    trainingback.Visible = true;
    trainingnext.Visible = true;
    eatinglabel.Visible = true;
    eatingactivity.Visible = true;
    dressinglabel.Visible = false;
    toiletinglabel1.Visible = false;
    sleepinglabel.Visible = false;
    eatingdescription.Visible = true;
    eatingactivity.Visible = true;
    dressingactivity.Visible = false;
    dressingdescription.Visible = false;
    toiletingdescription.Visible = false;
    toiletingactivity.Visible = false;
    sleepingdescription.Visible = false;
    sleepingactivity.Visible = false;
    this.eatingassessmentBindingSource.MoveFirst();
}
```

```
        private void toiletingrb_CheckedChanged(object sender, EventArgs e)
        {
            trainingwelcome.Visible = false;
            trainingback.Visible = true;
            trainingnext.Visible = true;
            eatinglabel.Visible = false;
            dressinglabel.Visible = false;
            toiletinglabel1.Visible = true;
            sleepinglabel.Visible = false;
            eatingdescription.Visible = false;
            eatingactivity.Visible = false;
            dressingdescription.Visible = false;
            dressingactivity.Visible = false;
            toiletingdescription.Visible = true;
            toiletingactivity.Visible = true;
            sleepingdescription.Visible = false;
            sleepingactivity.Visible = false;
            this.toiletingassessmentBindingSource.MoveFirst();
        }

        private void dressingrb_CheckedChanged(object sender, EventArgs e)
        {
            trainingwelcome.Visible = false;
            trainingback.Visible = true;
            trainingnext.Visible = true;
            eatinglabel.Visible = false;
            dressinglabel.Visible = true;
            toiletinglabel1.Visible = false;
            sleepinglabel.Visible = false;
            eatingdescription.Visible = false;
            eatingactivity.Visible = false;
            dressingdescription.Visible = true;
            dressingactivity.Visible = true;
            toiletingdescription.Visible = false;
            toiletingactivity.Visible = false;
            sleepingdescription.Visible = false;
            sleepingactivity.Visible = false;
            this.dressingassessmentBindingSource.MoveFirst();
        }

        private void eatingactivity_TextChanged(object sender, EventArgs e)
        {

        }
    }

}
```

# APPENDIX B

## Self-Help Assessment and Training Support System User Manual

# Self-Help Assessment and Training Support System User Manual

# TABLE OF CONTENT


| TITLE | PAGE |
|---|---|
| Product Overview | 4 |
| Term of Service | 6 |
| Privacy Policy | 9 |
| Installation | 10 |
| Getting Started | 11 |
| FAQ | 20 |

**Product Overview**

The Self-Help Assessment and Training Support System consist of 2 parts

a) Computer with monitor

b) Water level detection

Self-Help Assessment and Training Support System (SHATSS) is an Early Intervention Program (EIP) to help special children mainly children with Down syndrome as well as normal children on assessing, evaluating and improving self-help skills. SHATSS is develop based on the existing manual assessment which have been tested on normal and children with special needs. It is an easy to use system with interactive user friendly audio instruction to give command to the user when operating.

SHATSS can be carried out for both normal children and children with special needs. It can perform assessment, evaluation based on assessment and training to improve the child ability on self-help skills. User can perform 4 self-help skills assessment and training namely eating, dressing, toileting and sleeping. A chart is displayed after evaluation to see the child's progress.

The Self-help skills are show in details at the following table. Each activities under the groups in Language ability as according to Hawaii Early Learning Profile (Help) and Denver curriculum will be fully assessed and selective guidance for training will be given in the training part.

**Table 1:** Self-help skills element

| Domain | Subgroups | Number of activities |
|---|---|---|
| Self-Help | Eating | 35 |
| | Dressing | 15 |
| | Toileting | 11 |
| | Sleeping | 10 |

**Term of Service**

**1.1. Terms**

By accessing this web site, you are agreeing to be bound by these web site Terms and Conditions of Use, all applicable laws and regulations, and agree that you are responsible for compliance with any applicable local laws. If you do not agree with any of these terms, you are prohibited from using or accessing this site. The materials contained in this web site are protected by applicable copyright and trade mark law.

**1.2. Use License**

Permission is granted to temporarily download one copy of the materials (information or software) on SHATSS for personal, non-commercial transitory viewing only. This is the grant of a license, not a transfer of title, and under this license you may not:

**1.2.1.** Modify or copy the materials;

**1.2.2.** Use the materials for any commercial purpose, or for any public display (commercial or non-commercial);

**1.2.3.** Attempt to decompile or reverse engineer any software contained on SHATSS;

**1.2.4.** Remove any copyright or other proprietary notations from the materials; or

**1.2.5.** Transfer the materials to another person or "mirror" the materials on any other server.

**1.2.6.** This license shall automatically terminate if you violate any of these restrictions and may be terminated by SHATSS at any time. Upon terminating your viewing of these materials or upon the termination of this license, you must destroy any downloaded materials in your possession whether in electronic or printed format.

### 1.3. Disclaimer

The materials on SHATSS are provided "as is". SHATSS makes no warranties, expressed or implied, and hereby disclaims and negates all other warranties, including without limitation, implied warranties or conditions of merchantability, fitness for a particular purpose, or non-infringement of intellectual property or other violation of rights. Further, SHATSS does not warrant or make any representations concerning the accuracy, likely results, or reliability of the use of the materials on its Internet web site or otherwise relating to such materials or on any sites linked to this site.

### 1.4. Limitations

In no event shall SHATSS or its suppliers be liable for any damages (including, without limitation, damages for loss of data or profit, or due to business interruption,) arising out of the use or inability to use the materials on SHATSS, even if SHATSS or a SHATSS authorized representative has been notified orally or in writing of the possibility of such damage. Because some jurisdictions do not allow limitations on implied warranties, or limitations of liability for consequential or incidental damages, these limitations may not apply to you.

### 1.5. Revisions and Errata

The materials appearing on SHATSS could include technical, typographical, or photographic errors. SHATSS does not warrant that any of the materials on its web site are accurate, complete, or current. SHATSS may make changes to the materials contained on its web site at any time without notice. SHATSS does not, however, make any commitment to update the materials.

### 1.6. Links

SHATSS has not reviewed all of the sites linked to its Internet web site and is not responsible for the contents of any such linked site. The inclusion of any link does not imply endorsement by SHATSS of the site. Use of any such linked web site is at the user's own risk.

### 1.7. Terms of Use

SHATSS may revise these terms of use for its web site at any time without notice. By using this web site you are agreeing to be bound by the then current version of these Terms and Conditions of Use.

### 1.8. Governing Law

Any claim relating to SHATSS shall be governed by the laws of the State of Malaysia without regard to its conflict of law provisions.

**Private Policy**

Your privacy is very important to us. Accordingly, we have developed this Policy in order for you to understand how we collect, use, communicate and disclose and make use of personal information. The following outlines our privacy

- Before or at the time of collecting personal information, we will identify the purposes for which information is being collected.
- We will collect and use of personal information solely with the objective of fulfilling those purposes specified by us and for other compatible purposes, unless we obtain the consent of the individual concerned or as required by law
- We will only retain personal information as long as necessary for the fulfillment of those purposes.
- We will collect personal information by lawful and fair means and, where appropriate, with the knowledge or consent of the individual concerned.
- Personal data should be relevant to the purposes for which it is to be used, and, to the extent necessary for those purposes, should be accurate, complete, and up-to-date.
- We will protect personal information by reasonable security safeguards against loss or theft, as well as unauthorized access, disclosure, copying, use or modification.
- We will make readily available to customers information about our policies and practices relating to the management of personal information.

We are committed to conducting our business in accordance with these principles in order to ensure that the confidentiality of personal information is protected and maintained.

## Installation

With the help of simple instructions, the following section will guide user to install this software.

**1.1.** Install from CD

Place the LATSS CD into computer or laptop CD drive. Installation will be started with auto run setting. User can also browse the SHATSS folder “setup” file and double click “setup” to run the installation setup.

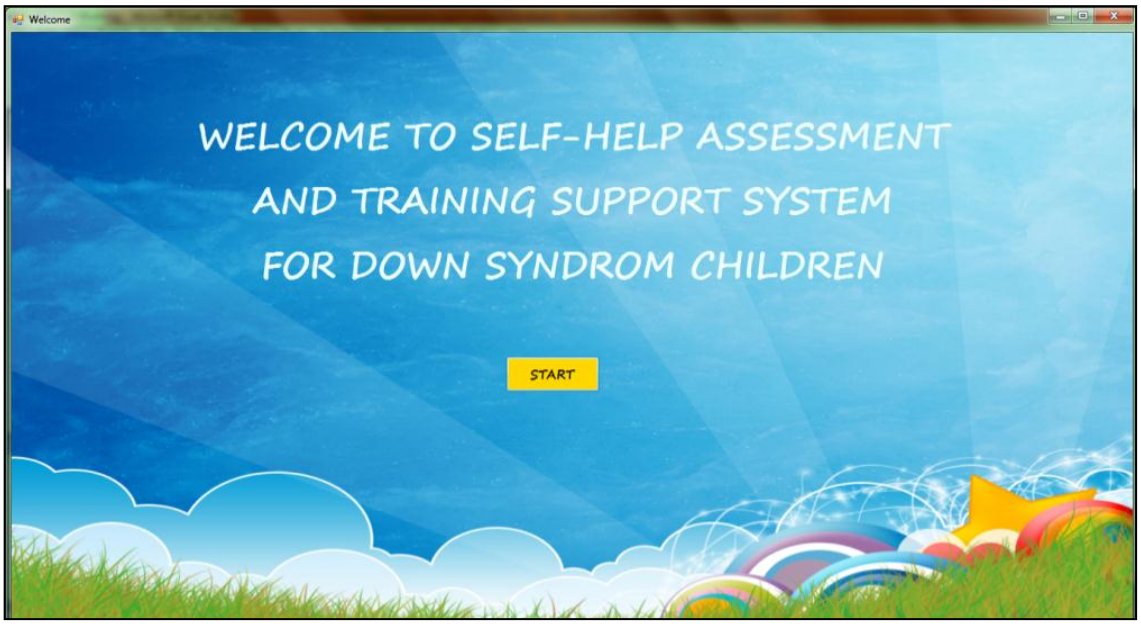
## Getting Started

After successful installation, user can now start operating SHATSS. Audio instruction is applied in the support system to help the user. User can listen to the instruction when operating SHATSS for guidance.

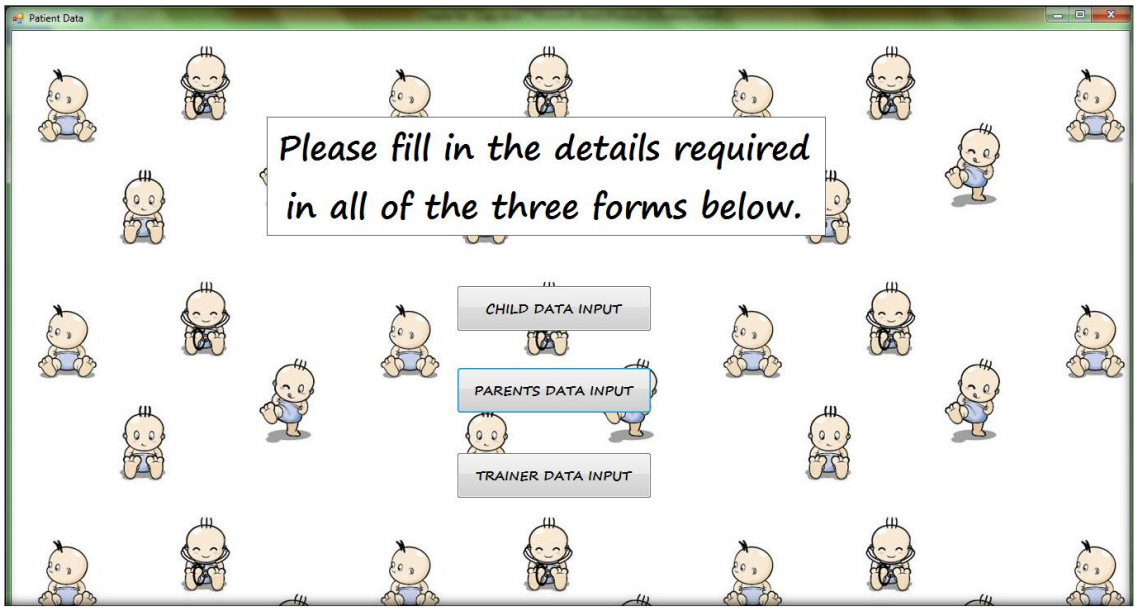
### 1.1 Start

Open SHATSS by clicking its icon. SHATSS window will open. Follow the instruction to get started using SHATSS.

1. Click “START” to begin.




2. Once “START” button is clicked, patient data form window will display. Fill all three forms; child data input, parents data input and trainer data input, by clicking the respective button.


Patient Data
Please fill in the details required in all of the three forms below.
CHILD DATA INPUT
PARENTS DATA INPUT
TRAINER DATA INPUT

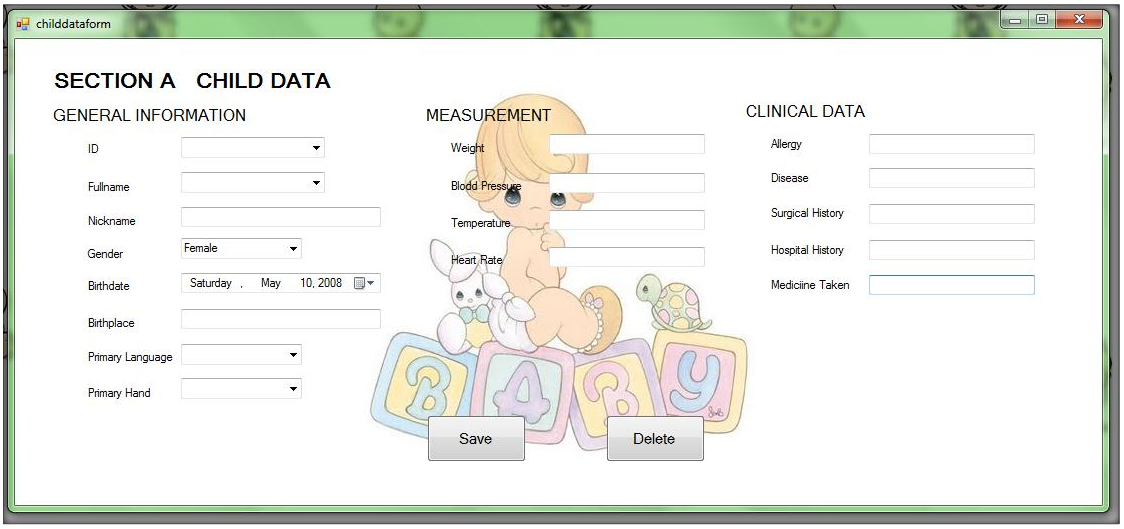
## 1.2 Registration

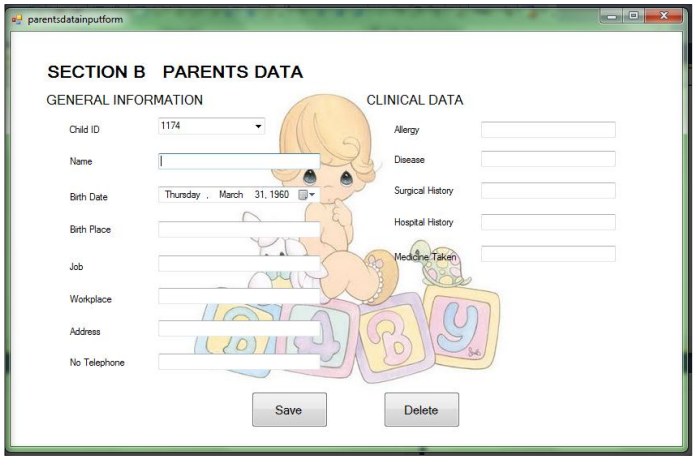
Clicking child data input, parents data input and trainer data input buttons respectively will display different forms. Fill in all the details required. The forms are displayed as follow.

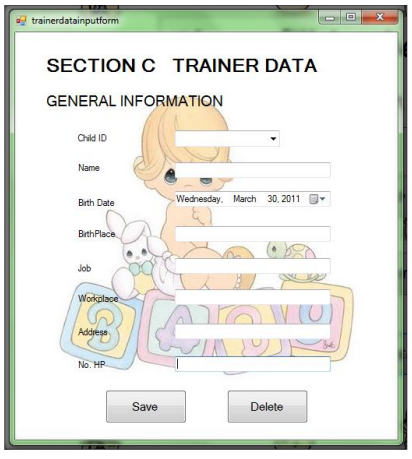
1. If the child is not registered with an ID, user can assign a new ID to the child by key in the ID at the “ID” field. Fill all the child data on the required field and click “SAVE” to save the information.
2. If the child is already registered with an ID, user can click the drop down button at the “ID” field to select the child’s ID. The information will be displayed upon clicking the child’s ID.
3. User can also display the child information by selecting the child’s full name at the “Fullname” field.
4. User can delete the existing child data by selecting the child ID or Fullname to display the selected child information and click “DELETE” button.

5. To change the existing child data, just rewrite the data on the desired text box and click “SAVE” button. The data is then updated with the new data.

The child data input, parents data input and trainer data input forms are shown below.



## 1.3 Assessment

After successfully register the child into the database, user can move on to assessment session. The assessment consists of 4 skills namely eating, dressing, toileting and sleeping.

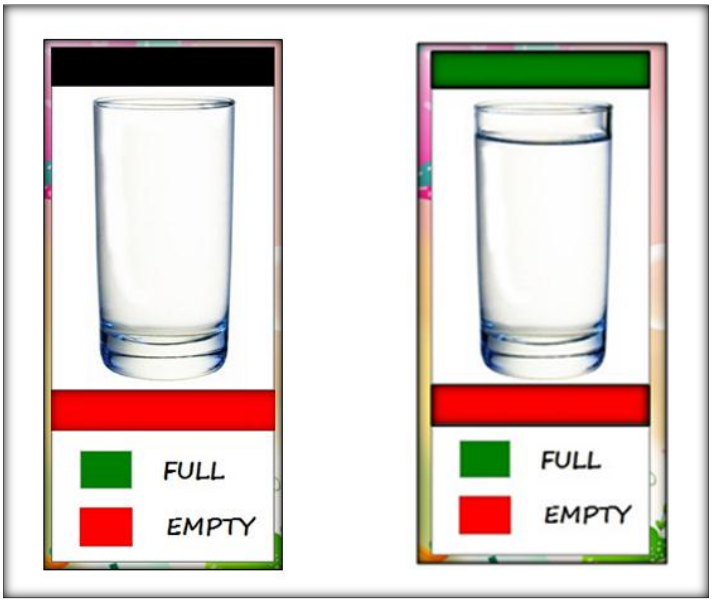
1. Click on the desired radio button skill that you want to perform with the child.
2. The skill's activity is display on the screen.
3. Evaluate the child based on the activity displayed. Click "Yes" button if the child can perform the specified activity and click "No" button if the child cannot perform the activity.
4. Finish evaluating all the activities of the selected skill. A message is display when the assessment is finish.
5. Then, user may select other skills to continue assessment.

The assessment form is shown below. User can close the assessment window and proceed to report section upon finishing the assessment.

Assessment using water level detection on the activity “pours liquid to the cup” in “Eating” skills.

1. Connect the RS232 USB converter cable to the computer.
2. Set the serial port of the USB cable to 4. Setting USB serial port can be change at *device manager*.
3. Open assessment window and select eating assessment. The first assessment in the eating list is the activity to pour liquid to the cup.
4. Ask the child to fill the cup with water.
5. When red LED lit on, it indicates that the cup is half full and the child requires to fill the cup with more water. When green LED lit on, this in turn indicates that the cup is full with water. The child is successfully undergo the assessment for pouring liquid into the cup.
6. User can proceed to do the next assessment.

The figure below shows the LED indication when the cup is half full and full condition.

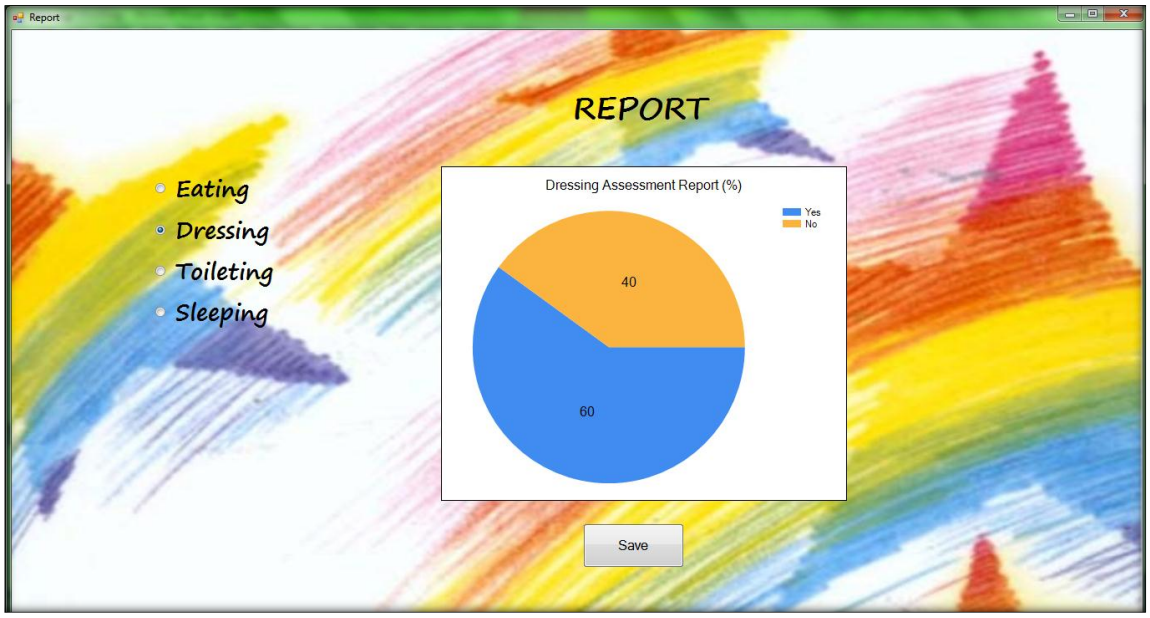


## 1.4 Report

User can check the child's progress in report section. A pie chart is display for all of the assessment that already been evaluated earlier.

1. Click the radio button skill to view the pie chart.
2. Click the "SAVE" button to save the report for record.

The target is to produce a higher percentage in the report. Higher percentage in the "Yes" field indicates that the child had mastered the skills activity. If the result is unsatisfying, user can proceed to training section to improve the child's self-help skill ability. The report is shown as below.



## 1.5 Training

Training section is for user to improve the child's self-help skill ability.

1. Click on the specified skills to perform the training on the child.
2. A list of instruction is displayed for training.
3. Follow the instruction to perform the training on the child.
4. Click "Next" button after the child successfully perform the activity. User can always return back to the previous activity by clicking "Back" button.
5. A message is display if the training is finish. User can proceed for other skill training.

The training window is shown below.

## FAQ

### 1.1. How to begin?

To use SHATSS with an installer CD, you should follow the instructions in the user manual to begin your child's/children's ability assessment.

### 1.2. What is the function of LATSS?

SHATSS is designed mainly to assess the self-help skills ability of the children. SHATSS can also be used as data management for children and information centre for parents or trainers.

### 1.3. Who is/are suitable to use/to be used this tool?

This tool is designed to be used by the public especially for parents, pre-school teachers or occupational therapists for children with special needs. However, SHATSS also can be used on normal child. LATSS can be used to assess the developmental abilities for special children under six years old and normal children under three years old.

### 1.4. Why should we use LATSS?

This tool is useful for parents or trainers to detect the strength and weaknesses of their children in the domains mentioned above. The database of this tool is highly reliable as it is based on the international recognized assessment method and suggestions based on experience trainers.

# APPENDIX C

## List of Self-Help Skills and List of Self-Help Subgroup Skills

**List of self-help skills**

| no | Action | Description |
|---|---|---|
| 1 | Responses to food stimulus | Child opens and closes mouth in response to food stimulus |
| 2 | Sucks and swallows reflex are inhibited | Childs is not able to do the actions of suck and swallow |
| 3 | Brings hand to mouth | Child is able to bring hand to the mouth independently |
| 4 | Recognizes bottle visually | Child is able to recognize bottle visually from other things |
| 5 | Coordinates sucking, swallowing, and breathing | Child is able to suck on bottle or nipple appropriately |
| 6 | Brings hand with toy or object to mouth | Child brings hand with toy or object to the mouth |
| 7 | Pats bottle | Child pats the bottle using hands |
| 8 | Places both hands on bottle | Child places both hands on the bottle appropriately |
| 9 | Swallows strained or pureed foods | Child swallows strained food which are fed to him/her |
| 10 | Rooting reflex is inhibited | Child is not able to perform rooting |
| 11 | Mouths and munches solid foods | Child mouths and gums solid food |
| 12 | Bites and chews toys | Child brings the toys to the mouth and bites them |
| 13 | Uses tongue during eating | Child uses tongue during eating and before the foods are swallowed |
| 14 | Feeds self a cracker | Child holds and feed self a cracker |
| 15 | Holds own bottle | Child holds own bottles independently |
| 16 | Bites food voluntarily | Child bites the foods voluntarily during eating |
| 17 | Drinks from cup held for him | Child is able to drink from cup held for him correctly |
| 18 | Drools less except when teething | Child drools less except when teething |
| 19 | Feeds self with fingers | Child feeds himself/herself using hands and fingers |
| 20 | Holds spoon | Child is able to hold the spoon independently |
| 21 | Chews food with coordinated movements | Child chews food with coordinated movements correctly |
| 22 | Brings spoon to mouth - turns spoon over | Child brings the spoon to his mouth and licks the foods left on the |

| | | |
|---|---|---|
| | | spoon |
| 23 | Holds cup handle | Child has better control of his/her hands and can hold the cup with his/her hands |
| 24 | May refuse foods - appetite decreases | When the appetite decreases, the child may refuse to eat |
| 25 | Holds and drinks from cup with some spilling | Child holds and drinks from the cup with some spilling |
| 26 | Gives empty dish to adult | Child finishes the foods and gives empty dish to the adults |
| 27 | Distinguishes between edible and inedible objects | Child knows what can be eaten and cannot be eaten |
| 28 | Plays with food | Child may play with the foods given |
| 29 | Scoops food, feeds self with spoon with some spilling | Child is able to scoop the food, feed self with spoon with some spilling |
| 30 | Chews completely with rotary jaw movements | Child chews completely with rotary jaw movements while eating |
| 31 | Handles bottle to adults | Child handles the bottle to adult independently |
| 32 | May have definite food preferences | Child shows self preference on the food |
| 33 | Unwraps food | Child removes the wrapper of the food and eats the food |
| 34 | Pours liquid from small container | Child pours liquid from a small container independently |
| 35 | Uses fork | Child is able to use fork to eat and knows how to use it |
| 36 | Uses napkin | Child is able to use napkin after eating and knows how to use it |
| 37 | Serves self at table with little spilling | Child serves self at the table during eating independently with little spilling |
| 38 | Shows interest in setting table | Child may show interest in the setting of the table |
| 39 | Cooperates with dressing by coordinating arm or leg | Child voluntarily moves his/her arms or legs while being dressed |
| 40 | Removes hat | Child is able to remove the hat off his/her head |
| 41 | Removes socks | Child is able to remove socks off his/her feet |
| 42 | Places hat on head | Child is able to place the hat on his/her head |
| 43 | Unzips, zips large zipper | Child is able to zip and unzip dress and pants |

| | | |
|---|---|---|
| 44 | Removes shoes | Child takes off shoes by himself/herself |
| 45 | Unbuttons large buttons | Child unbuttons large buttons independently |
| 46 | Pulls pants down with assistance | Child pulls his/her pants down with assistance over the hips |
| 47 | Pulls pants up with assistance | Child pulls his/her pants up with assistance over the hips |
| 48 | Puts shoes on with assistance | Child puts shoes on his/her feet with assistance |
| 49 | Undresses with assistance | Child needs assistance during undressing himself/herself |
| 50 | Dresses self with assistance | Child dresses himself/herself with assistance |
| 51 | Buttons large buttons | Child buttons large round buttons |
| 52 | Hangs clothing on hook | Child knows where to hang clothes |
| 53 | Dresses with supervision, requires assistance with fastenings | Child is able to dress with supervision and help is needed during fastening |
| 55 | Indicates discomfort over soiled pants | Child shows discomfort over dirty pants with soil |
| 56 | Sits on potty chair or on adaptive seat in toilet with assistance | Child sits on potty chair or on adaptive seat in the toilet with assistance |
| 57 | May go to toilet regulated by adult | Child may go to toilet according to the regulation by adult |
| 58 | Anticipates need to eliminate - uses same word for both functions | Child anticipates need to eliminate |
| 59 | Anticipates need to eliminate in time | Child anticipates need to eliminate in time |
| 60 | Uses toilet with assistance - has daytime control | Child uses toilet with assistance |
| 61 | Distinguishes between urination and bowel movements | Child is able to distinguish between urination and bowel movements |
| 62 | Verbalizes need to go to toilet - has occasional accidents | Child verbalizes need to go to toilet |
| 63 | Takes responsibility for toileting | Child knows the responsibility for toileting |
| 64 | Washes and dries hands partially | Child partially washes and dries his/her hands; child requires assistance to complete the action |
| 65 | Washes hands | Child washes his/her hands by himself/herself |
| 66 | Wipes nose with assistance | Child wipes his/her nose with some help |

| | | |
|---|---|---|
| 67 | Blows nose with assistance | Child blows nose with help |
| 68 | Dries hands | Child dries his/her own hands with a towel |
| 69 | Brushes teeth with assistance | Child begins to brush teeth with assistance |
| 70 | Helps with bathing self | Child is able to bathe himself/herself |
| 71 | Sleeps ten-fifteen hours daily | Child sleeps ten-fifteen hours daily |
| 72 | May awake and cry from dreams | Child may awake and cry from his/her dreams |
| 73 | May refuse naps | Child may refuse naps sometimes |
| 74 | Sleeps four-ten hour intervally | Child sleeps four-ten hour with intervals at night |
| 75 | Stays awake for long periods without crying at night | Child stays awake for long periods without crying at night |
| 76 | Naps frequently | Child naps frequently in a day |
| 77 | Sleeps ten-twelve hours with night awakening | Child sleeps ten-twelve hours with night awakening |
| 78 | Naps one-four hours, two-three times each day | Child naps one-four hours. two-three times each day |
| 79 | Sleeps twelve to fourteen hours | Child sleeps twelve to fourteen hours each day |
| 80 | Naps one-four hours, once or twice each day; may refuse morning nap | Child naps one-four hours, once or twice each day; child may refuse napping in the morning |
| 81 | Sleeps ten-twelve hours | Child sleeps ten-twelve hours a day |
| 82 | Naps one-three hours once in afternoon | Child naps around one-three hours once in the afternoon |
| 83 | Opens doors by turning knob | Child opens doors by turning knob independently |
| 84 | Helps with simple household tasks | Child is able to help with simple household tasks |
| 85 | Handles fragile items carefully | Child is aware and handles fragile items carefully |
| 86 | Helps put things away | Child voluntarily helps to put things away |
| 87 | Holds small cup in one hand | Child is able to hold small cup in one hand appropriately |
| 88 | Understands and stays away from common dangers - stairs, glass, strange animals | Child understands and stays away from common dangers such as stairs, glass, and strange animals |
| 89 | Holds spoon in hands - palm up | Child holds spoon in hand with palm up |
| 90 | Delays sleeping by demanding things | Child tends to delay sleeping by demanding things |

| | | |
|---|---|---|
| 91 | May reject many foods | Child may reject many types of foods |
| 92 | Knows proper place for own things | Child knows the proper location to place for own things |
| 93 | Insists on doing things independently | Child demonstrates determination and insists on doing things independently |

**Self-help subgroup skills**

| **Eating skill** | | |
|---|---|---|
| **no** | **Action** | **Description** |
| 1 | Responses to food stimulus | Child opens and closes mouth in response to food stimulus |
| 2 | Sucks and swallows reflex are inhibited | Childs is not able to do the actions of suck and swallow |
| 3 | Brings hand to mouth | Child is able to bring hand to the mouth independently |
| 4 | Recognizes bottle visually | Child is able to recognize bottle visually from other things |
| 5 | Coordinates sucking, swallowing, and breathing | Child is able to suck on bottle or nipple appropriately |
| 6 | Brings hand with toy or object to mouth | Child brings hand with toy or object to the mouth |
| 7 | Pats bottle | Child pats the bottle using hands |
| 8 | Places both hands on bottle | Child places both hands on the bottle appropriately |
| 9 | Swallows strained or pureed foods | Child swallows strained food which are fed to him/her |
| 10 | Rooting reflex is inhibited | Child is not able to perform rooting |
| 11 | Mouths and munches solid foods | Child mouths and gums solid food |
| 12 | Bites and chews toys | Child brings the toys to the mouth and bites them |
| 13 | Uses tongue during eating | Child uses tongue during eating and before the foods are swallowed |
| 14 | Feeds self a cracker | Child holds and feed self a cracker |

| | | |
|---|---|---|
| 15 | Holds own bottle | Child holds own bottles independently |
| 16 | Bites food voluntarily | Child bites the foods voluntarily during eating |
| 17 | Drinks from cup held for him | Child is able to drink from cup held for him correctly |
| 18 | Drools less except when teething | Child drools less except when teething |
| 19 | Feeds self with fingers | Child feeds himself/herself using hands and fingers |
| 20 | Holds spoon | Child is able to hold the spoon independently |
| 21 | Chews food with coordinated movements | Child chews food with coordinated movements correctly |
| 22 | Brings spoon to mouth - turns spoon over | Child brings the spoon to his mouth and licks the foods left on the spoon |
| 23 | Holds cup handle | Child has better control of his/her hands and can hold the cup with his/her hands |
| 24 | May refuse foods - appetite decreases | When the appetite decreases, the child may refuse to eat |
| 25 | Holds and drinks from cup with some spilling | Child holds and drinks from the cup with some spilling |
| 26 | Gives empty dish to adult | Child finishes the foods and gives empty dish to the adults |
| 27 | Distinguishes between edible and inedible objects | Child knows what can be eaten and cannot be eaten |
| 28 | Plays with food | Child may play with the foods given |
| 29 | Scoops food, feeds self with spoon with some spilling | Child is able to scoop the food, feed self with spoon with some spilling |
| 30 | Chews completely with rotary jaw movements | Child chews completely with rotary jaw movements while eating |
| 31 | Handles bottle to adults | Child handles the bottle to adult independently |
| 32 | May have definite food preferences | Child shows self preference on the food |
| 33 | Unwraps food | Child removes the wrapper of the food and eats the food |
| 34 | Pours liquid from small container | Child pours liquid from a small container independently |
| 35 | Uses fork | Child is able to use fork to eat and knows how to use it |
| 36 | Uses napkin | Child is able to use napkin after eating and knows how to use it |
| 37 | Serves self at table with little spilling | Child serves self at the table during eating independently with little spilling |

| | | |
|---|---|---|
| 38 | Shows interest in setting table | Child may show interest in the setting of the table |
| 39 | Helps with simple household tasks | Child is able to help with simple household tasks |
| 40 | Handles fragile items carefully | Child is aware and handles fragile items carefully |
| 41 | Helps put things away | Child voluntarily helps to put things away |
| 42 | Holds small cup in one hand | Child is able to hold small cup in one hand appropriately |
| 43 | Understands and stays away from common dangers - stairs, glass, strange animals | Child understands and stays away from common dangers such as stairs, glass, and strange animals |
| 44 | Holds spoon in hands - palm up | Child holds spoon in hand with palm up |
| 45 | May reject many foods | Child may reject many types of foods |
| 46 | Knows proper place for own things | Child knows the proper location to place for own things |
| 47 | Insists on doing things independently | Child demonstrates determination and insists on doing things independently |
| 48 | Insists on doing things independently | Child demonstrates determination and insists on doing things independently |

| Dressing | | |
|---|---|---|
| **No** | **Action** | **Description** |
| 1 | Cooperates with dressing by coordinating arm or leg | Child voluntarily moves his/her arms or legs while being dressed |
| 2 | Removes hat | Child is able to remove the hat off his/her head |
| 3 | Removes socks | Child is able to remove socks off his/her feet |
| 4 | Places hat on head | Child is able to place the hat on his/her head |
| 5 | Unzips, zips large zipper | Child is able to zip and unzip dress and pants |
| 6 | Removes shoes | Child takes off shoes by himself/herself |
| 7 | Unbuttons large buttons | Child unbuttons large buttons independently |
| 8 | Pulls pants down with assistance | Child pulls his/her pants down with assistance over the hips |
| 9 | Pulls pants up with assistance | Child pulls his/her pants up with assistance over the hips |
| 10 | Puts shoes on with assistance | Child puts shoes on his/her feet with assistance |
| 11 | Undresses with assistance | Child needs assistance during undressing himself/herself |
| 12 | Dresses self with assistance | Child dresses himself/herself with assistance |
| 13 | Buttons large buttons | Child buttons large round buttons |
| 14 | Hangs clothing on hook | Child knows where to hang clothes |
| 15 | Dresses with supervision, requires assistance with fastenings | Child is able to dress with supervision and help is needed during fastening |
| 16 | Insists on doing things independently | Child demonstrates determination and insists on doing things independently |

| Toileting | | |
|---|---|---|
| **No** | **Action** | **Description** |
| 1 | Indicates discomfort over soiled pants | Child shows discomfort over dirty pants with soil |
| 2 | Sits on potty chair or on adaptive seat in toilet with assistance | Child sits on potty chair or on adaptive seat in the toilet with assistance |
| 3 | May go to toilet regulated by adult | Child may go to toilet according to the regulation by adult |
| 4 | Anticipates need to eliminate - uses same word for both functions | Child anticipates need to eliminate |
| 5 | Anticipates need to eliminate in time | Child anticipates need to eliminate in time |
| 6 | Uses toilet with assistance - has daytime control | Child uses toilet with assistance |
| 7 | Distinguishes between urination and bowel movements | Child is able to distinguish between urination and bowel movements |
| 8 | Verbalizes need to go to toilet - has occasional accidents | Child verbalizes need to go to toilet |
| 9 | Takes responsibility for toileting | Child knows the responsibility for toileting |
| 10 | Washes and dries hands partially | Child partially washes and dries his/her hands; child requires assistance to complete the action |
| 11 | Washes hands | Child washes his/her hands by himself/herself |
| 12 | Wipes nose with assistance | Child wipes his/her nose with some help |
| 13 | Blows nose with assistance | Child blows nose with help |
| 14 | Dries hands | Child dries his/her own hands with a towel |
| 15 | Brushes teeth with assistance | Child begins to brush teeth with assistance |
| 16 | Helps with bathing self | Child is able to bathe himself/herself |
| 17 | Opens doors by turning knob | Child opens doors by turning knob independently |
| 18 | Understands and stays away from common dangers - stairs, glass, strange animals | Child understands and stays away from common dangers such as stairs, glass, and strange animals |
| 19 | Knows proper place for own things | Child knows the proper location to place for own things |

| | | |
|---|---|---|
| 20 | Insists on doing things independently | Child demonstrates determination and insists on doing things independently |

| **Sleeping** | | |
|---|---|---|
| **No** | **Action** | **Description** |
| 1 | Sleeps ten-fifteen hours daily | Child sleeps ten-fifteen hours daily |
| 2 | May awake and cry from dreams | Child may awake and cry from his/her dreams |
| 3 | May refuse naps | Child may refuse naps sometimes |
| 4 | Sleeps four-ten hour intervally | Child sleeps four-ten hour with intervals at night |
| 5 | Stays awake for long periods without crying at night | Child stays awake for long periods without crying at night |
| 6 | Naps frequently | Child naps frequently in a day |
| 7 | Sleeps ten-twelve hours with night awakening | Child sleeps ten-twelve hours with night awakening |
| 8 | Naps one-four hours, two-three times each day | Child naps one-four hours. two-three times each day |
| 9 | Sleeps twelve to fourteen hours | Child sleeps twelve to fourteen hours each day |
| 10 | Naps one-four hours, once or twice each day; may refuse morning nap | Child naps one-four hours, once or twice each day; child may refuse napping in the morning |
| 11 | Sleeps ten-twelve hours | Child sleeps ten-twelve hours a day |
| 12 | Naps one-three hours once in afternoon | Child naps around one-three hours once in the afternoon |
| 13 | Delays sleeping by demanding things | Child tends to delay sleeping by demanding things |